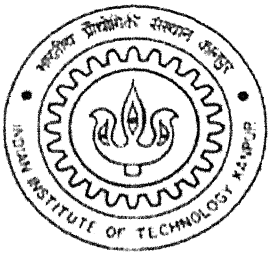
Improved Methodologies for Rainfall-Runoff Modeling using Conceptual and Soft Computing Techniques and Exploration of Physical Significance in ANN Models

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

by

Sanaga Srinivasulu

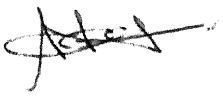


DEPARTMENT OF CIVIL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

July 2003

CERTIFICATE

It is certified that the work contained in the thesis entitled: ***Improved Methodologies for Rainfall-Runoff Modeling using Conceptual and Soft Computing Techniques and Exploration of Physical Significance in ANN Models by*** **Sanaga Srinivasulu** has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.



Signature of Supervisor
Dr. Ashu Jain
Department of Civil Engineering
Indian Institute of Technology
Kanpur

July 2003

SYNOPSIS

Modeling of the rainfall-runoff process has been the subject of research among hydrologists and engineers for a very long time. The transformation of rainfall into runoff is an extremely complex, dynamic, and non-linear process which is affected by many factors, such as storm, watershed, geo-morphological, and climatic characteristics, which are often inter-related. The influence of these factors and many of their combinations in generating runoff is an extremely complex physical process and is not understood clearly. The approaches adopted for modeling the complex, dynamic, and non-linear rainfall-runoff process cover a wide range of methods from completely black-box models to very detailed deterministic models. The deterministic models of the rainfall-runoff process use equations of mass, energy, and momentum to describe the movement of water over land surface, and through the unsaturated and saturated zones of the earth. The deterministic models consist of very complex structures and their development requires a lot of experience and expertise. Hydrologists have responded to these limitations by developing conceptual rainfall-runoff (CRR) models, in which, instead of using the equations of mass, energy, and momentum to describe the process of water movement, a simplified, but plausible conceptual representation of underlying physics is adopted. The systems-theoretic models, also known as the black box models, attempt to develop relationships among input and output variables involved in a physical process without considering the underlying physical process. Conventional stochastic models (ARIMA type) and the Artificial Neural Network (ANN) models fall under the systems-theoretic category of rainfall-runoff models.

There are certain issues that have either been ignored completely or not explored fully by researchers and hydrologists in the past while developing rainfall-runoff models. These include: **(a) Integration of Techniques:** Historically, researchers have employed the techniques available for rainfall-runoff modeling, i.e. deterministic or systems-theoretic, in isolation. However, it may be possible to achieve better model performance if the non-linear systems-theoretic models are developed with some conceptual component embedded in them. Such an integrated approach will result in a new class of models called *grey-box* models that are able to take advantage of both deterministic and systems-theoretic techniques. **(b) Modeling Decomposed Flow Hydrograph:** The runoff response of a watershed, represented by different segments of a flow hydrograph, is produced by different physical processes ongoing in a watershed. The rising and falling limbs, and their different portions, result from different physical processes having different dynamics influenced by different hydrological and climatic factors. It has also been reported in literature that the rainfall-runoff mapping in a watershed can be fragmented or discontinuous having significant variations over the input space because of the functional relationships between rainfall and runoff being quite different for low, medium, and high magnitudes of streamflow. In order to capture such fragmented and/or discontinuous relationships, one first needs to decompose the complex rainfall-runoff mapping problem into several simple problems, each of which can then be tackled easily and more effectively. **(c) Training of ANN Rainfall-Runoff Models:** Most of the ANN applications reported for rainfall-runoff modeling have employed the back-propagation (BP) training algorithm and its variations. Many such studies have reported that the ANN rainfall-runoff models trained using BP algorithm and its variations have not been able to train the low magnitude flows properly and therefore result in poor overall generalized

rainfall-runoff relationships. A strong need to attempt different training methods based on the combination of deterministic and probabilistic approaches has been emphasized. **(d) Physics inside Trained ANN Models:** The ANN models of a physical process are considered to be black boxes. However, it must be realized that if the nonlinear function being mapped by the ANN is explored further, one may be able to shed some light into the physical processes of the system being modeled that are inherent in a trained ANN. It is quite possible that while training an ANN, different hidden neurons do learn the individual relationships inherent in different portions of a flow hydrograph. However, identifying components of a physical process in a trained ANN is an area of research that is more or less virgin and remains unexplored, especially in hydrology.

The objectives of the present research effort are to: (i) develop CRR models that are capable of capturing the complex dynamic and non-linear rainfall-runoff process in a large watershed, (ii) explore the use of *real-coded* genetic algorithm (RGA) to calibrate parameters of the CRR models, (iii) develop black box type ANN (called BANN) models of the rainfall-runoff process, and compare their performances with the CRR models, (iv) explore the possibility of developing ANN rainfall-runoff models with conceptual component(s) embedded in them (called *grey-box* ANN models or GANN models), and compare their performances with the BANN models, (v) explore the use of RGA to train both BANN and GANN rainfall-runoff models, (vi) investigate the validity of the hypothesis that the use of either soft or physics based heuristic decomposition of the input output data, and the development of rainfall-runoff models for each decomposed class corresponds to different dynamics of the rainfall-runoff process, and therefore, results in better overall model performance, and (vii) investigate for the presence of physics inside a

trained ANN rainfall-runoff model by exploring the parallel distributed components of the ANN model in relation to the deterministic components of the hydrologic process. The daily total rainfall (mm) and daily average streamflow (m^3/s) derived from the Kentucky River watershed, USA were employed to develop various models investigated and the methodologies proposed in this study.

The following conclusions are drawn based on the results obtained in the study carried out in the thesis: (i) The performance of the CRR models was only reasonable, and that of the BANN models was found to be better than the CRR models, (ii) Integration of conceptual components in the BANN models marginally improved the performance of the resulting GANN models, (iii) The new technique of RGA can be used to calibrate CRR models effectively using continuous rainfall and streamflow data, (iv) The modeling of high and medium magnitude flows was found to be very good from all the ANN models trained using BP algorithm, and the use of RGA significantly improved the performance in modeling low magnitude flows and hence resulted in better overall generalized rainfall-runoff relationships, (v) Heuristic decomposition of the input output data based on physical processes and the use of integrated techniques (GDANN models) resulted in the best model performance among all the models investigated in this study, (vi) Decomposition of the input output data into different classes using Self Organizing Map (SOM) networks coupled with ANN models supports the concept that different dynamical processes inherent in different data sets belonging to different classes should be modeled separately to get better model performance, (vii) The study to investigate for the physics inherent in the trained ANN rainfall-runoff models revealed that the four neurons in the hidden layer of the BANN model represented base flow, delayed surface flow, interflow, and the rising limb,

respectively. The evidence of physics inside the trained ANN rainfall-runoff models found in this study can potentially initiate a long drawn debate on whether the ANN models are black boxes.

The results obtained in this study are preliminary in nature as they are based on the application of the proposed methodologies to a single watershed, and in order to validate the concepts and methodologies proposed, more studies need to be carried out by applying the proposed methodologies to other watersheds of varying hydrologic and climatic conditions. Further, the models developed and the data employed in this study were lumped in nature. Ideally, distributed hydrologic models need to be developed and explored to have more confidence in the findings of the studies such as the one carried out. It is hoped that future research will focus on these directions to strengthen the findings of this study.

TO

MY PARENTS

&

TEACHERS

ACKNOWLEDEGEMENTS

I have deep sense of gratitude to my respected supervisor Dr. Ashu Jain for his guidance and encouragement throughout my stay at IIT Kanpur. I am highly grateful to Prof. T. Gangadharaiah, who encouraged me to pursue my PhD at IIT Kanpur and also for his valuable support throughout the course of the work. I am extremely thankful to my respected teachers in the Hydraulics and Water Resource division of Civil Engineering Department, IIT Kanpur, Prof. Bithin Datta, and Dr. Rajesh Srivastava for their constant support. I wish to thank Dr. K.P. Sudheer, Department of Civil Engineering, IIT, Madras (Ex. Visiting Faculty at IIT Kanpur) for his valuable suggestions, invaluable time and support to my work, during his short stay at IIT Kanpur. I am sincerely thankful to Prof. Kalyanmoy Deb, Department of Mechanical Engineering, IIT Kanpur, for his valuable suggestions during and after the course work. I extend my gratitude to Prof. M.R. Madhav, Department of Civil Engineering for giving me invaluable suggestions and moral support during all stages of my work. I thank to our Vice-chancellor, Registrar of JNT University and to my colleagues and friends at JNTU Kakinada, Prof P. Venkata Narasaiah, Prof. A. Srirama Rao, Dr. P. Udya Bhaskar, Dr. Prasada Raju, and Dr K. Ramu for their encouragement to pursue my PhD at IIT Kanpur.

I wish to thank Ravinder, Saradhi and Sarvan, Sahu, Sudhakar, and Manohar, for their help during preparation of my thesis. I wish to give special thanks to Dr. Ganguly, who keeps my family and me in a good health. I express gratitude to Kaushik, Tiwari, Ramireddy, Siva Prasad, and Eeswari for their good company at SBRA and make my stay more delightful. I am also thankful to my fellow research scholars Sarat and Priti for their good

company at the Institute. And my sincere thanks to Satyanarayana and Rajib, who motivated me with their hard work. Thanks are also due to all Andhra student friends from Hall–IV, Hall-V, and RA Hostel, who engaged me and provided me a good company in the sports and other activities during these three years of stay at IIT Kanpur.

Last, but not least I tender my deep sense of regards to my wife Sridevi and to my sons Rohit and Pardhu and all my family members for their encouragement for utmost perseverance.

Above all, I bow my head before the Almighty, for all the blessings showered on me.

(Sanaga Srinivasulu)

CONTENTS

CERTIFICATE ii

SYNOPSIS iii

ACKNOWLEDGEMENTS ix

CONTENTS xi

LIST OF FIGURES xv

LIST OF TABLES xix

LIST OF SYMBOLS xx

CHAPTER 1 INTRODUCTION 1

1.1 General 1

1.2 Rainfall Runoff Modeling 2

1.3 Certain Issues in Rainfall Runoff Modeling 6

1.3.1 Issue of Calibration of Rainfall-Runoff Models 6

1.3.2 Issue of Integrated (or Grey Box) Rainfall-Runoff Models 7

1.3.3 Issue of Modeling Decomposed Flow Hydrograph 9

1.3.4 Issue of Training of ANN Rainfall-Runoff Models 10

1.3.5 Issue of Modeling Low Magnitude Flows 13

1.3.6 Issue of Physics inside a Trained ANN Model 14

1.4 Objectives of the Thesis 15

1.5 Organization of the Thesis 17

CHAPTER 2	**REVIEW OF LITERATURE**	18
	2.1 General	18
	2.2 Conceptual Rainfall-Runoff Models	19
	2.2.1 Development and Application of CRR Models	19
	2.2.2 Calibration of CRR Models	23
	2.3 ANNs in Hydrological Modeling	27
	2.3.1 Development and Application of ANN Models	27
	2.3.2 Training Algorithms for ANN Models	31
CHAPTER 3	**SOFT COMPUTING TECHNIQUES**	35
	3.1 General	35
	3.2 Artificial Neural Networks	36
	3.2.1 Multilayer Feed-Forward Neural Networks	37
	3.2.2 Self-Organizing Maps	42
	3.3 Genetic Algorithm	47
	3.3.1 GA Operators	48
	3.3.2 Advantages of *Real-coded* GA	54
	3.3.3 Elitism in *Real-coded* GA	57
CHAPTER 4	**DEVELOPMENT OF CRR MODELS**	59
	4.1 General	59
	4.2 CRR Model-I	61
	4.2.1 Base flow Component	64
	4.2.2 Infiltration Component	65
	4.2.3 Soil Moisture Accounting (SMA) Component	67
	4.2.4 Evapotranspiration Component	69

4.2.5 Surface Flow Component 69

4.3 CRR Model - II 74

4.3.1 Surface Flow Component 75

4.3.1.1 Linear Channel Component 75

4.3.1.1 Non-linear Reservoir Component 78

4.4 Study Area and Data 85

4.5 Calibration of CRR Models 87

4.5.1 Validation of *Elitist Real-Coded* GA 88

4.5.1 Calibration of Green–Ampt Infiltration Parameters 94

4.5.3 Performance Evaluation Indices 96

4.6 Discussion of Results from CRR Models 102

4.7 Summary 105

CHAPTER 5 DEVELOPMENT OF ANN RAINFALL-RUNOFF MODELS 110

5.1 General 110

5.2 Black-Box ANN (BANN) Models 111

5.2.1 Input Vector Identification for BANN Models 113

5.2.2 Training of BANN Models using *Elitist* RGA 117

5.2.3 Discussion of Results from BANN Models 119

5.2.3.1 Results from BANN Models 119

5.2.3.2 Comparison of BANN and CRR Models 132

5.3 *Grey-Box* ANN (GANN) Models 135

5.3.1 Discussion of Results from GANN Models 142

5.3.1.1 Results from GANN Models 142

5.3.1.2 Comparison of BANN Models and GANN Models 148

5.4 *Grey-Box* Decomposed (GDANN) Models 150

5.4.1 GDANN Models based on Heuristic Decomposition 155

5.4.2 GDANN Models based on Soft Decomposition 157

5.4.3 Discussion of Results from GDANN Models 161

5.5 Summary 172

CHAPTER 6 EXPLORATION OF PHYSICAL SIGNIFICANCE IN ANN MODELS 176

6.1 General 176

6.2 Analysis of Exploration of Physical Significance in ANN Models 179

6.2.1 Hidden Neurons with Observed Input Variables 181

6.2.2 Hidden Neurons with Conceptual Components of Hydrologic Process 183

CHAPTER 7 SUMMARY, CONCLUSIONS, LIMITATIONS 192

7.1 Summary 192

7.2 Conclusions 194

7.3 Limitations and Future Scope of the Study 198

REFERENCES 202

BIBILOGRAPHY 213

APPENDICES 215

LIST OF PUBLICATIONS 218

LIST OF FIGURES

Figure	No	Caption	Page No.
Figure	3.1	Structure of a Typical Feed-Forward Neural Network	38
Figure	3.2	Two-Dimensional Data Mapped onto One-Dimensional Space	44
Figure	3.3	Single Point Crossover in Binary Coded Genetic Algorithm	50
Figure	3.4	Flow Chart of Elitist *Real-Coded* GA	58
Figure	4.1	Schematic of the Simplified Hydrologic System	62
Figure	4.2	Schematic of the CRR Model –I	63
Figure	4.3	Base flow Model Component in CRR Models	65
Figure	4.4	Schematic of the CRR Model -I	76
Figure	4.5	A Non-linear Water Tank	79
Figure	4.6	Kentucky River Basin	86
Figure	4.7	Comparative Performance for Rastrigin Function	91
Figure	4.8	Comparative Performance for Six Hump Camel Back Function	91
Figure	4.9	Comparative Performance for Hartman Function	93
Figure	4.10	Comparative Performance Griewank Function	93
Figure	4.11a	Observed and Predicted Flows during Validation – From CRR Model -I	106
Figure	4.11b	Observed and Predicted Flows during Validation Period – From CRR Model-II	106
Figure	4.12a	Observed and Predicted Flows from CRR Model –II – From the Year 1986	107

Figure	4.12b	Observed and Predicted Flows from CRR Model –II – From the Year 1986	107
Figure	5.1	Cross Correlation of Rainfall with the Runoff at Time t at different lags	114
Figure	5.2	Autocorrelation Function of Rainfall Series	114
Figure	5.3	Autocorrelation Function of Runoff Series.	116
Figure	5.4	Partial Autocorrelation Function of Runoff Series	116
Figure	5.5	Flow Chart for Training ANN using *Elitist Real-Coded* GA	120
Figure	5.6(a)	Observed and Predicted Flows during Testing - From BANN-BP Model	123
Figure	5.6(b)	Observed and Predicted Flows during Testing - From BANN-GA Model	123
Figure	5.7(a)	Observed and Predicted Flows in the Dry Year 1986 - From BANN-BP Model	124
Figure	5.7(b)	Observed and Predicted Flows in the Dry Year 1986 - From BANN-GA Model	124
Figure	5.8(a)	Observed and Predicted Flows in the Wet Year 1989- From BANN-BP Model	125
Figure	5.8(b)	Observed and Predicted Flows in the Wet Year 1989 - From BANN-GA Model	125
Figure	5.9(a)	Observed and Predicted Low Flows in 1986 – From BANN-BP Model	126
Figure	5.9(b)	Observed and Predicted Low Flows in 1986 – From BANN-GA Model	126
Figure	5.10(a)	Transparent Box Model	136
Figure	5.10(b)	Black-Box Model	136
Figure	5.10(c)	Grey-Box Model	137
Figure	5.11	Structure of Grey-Box ANN Rainfall-Runoff Model	140

Figure	5.12	Cross Correlation between flow at Time t and Lagged Effective Rainfalls	141
Figure	5.13	Autocorrelation Function of Effective Rainfall Series	141
Figure	5.14(a)	Observed and Predicted Flows during Testing - From GANN-BP Model	144
Figure	5.14(b)	Observed and Predicted Flows during Testing - From GANN-GA Model	144
Figure	5.15(a)	Observed and Predicted Flows in the Dry Year 1986 - From GANN-BP Model	145
Figure	5.15(b)	Observed and Predicted Flows in the Dry Year 1986 - From GANN-GA Model	145
Figure	5.16(a)	Observed and Predicted Flows in the Wet Year 1989- From GANN-BP Model	146
Figure	5.16(b)	Observed and Predicted Flows in the Wet Year 1989 - From GANN-GA Model	146
Figure	5.17(a)	Observed and Predicted Low Flows in 1986 – From GANN-BP Model	147
Figure	5.17(b)	Observed and Predicted Low Flows in 1986 – From GANN-GA Model	147
Figure	5.18	Decomposition of a Flow Hydrograph	153
Figure	5.19	Dividing the Falling Limb	158
Figure	5.20	Dividing the Rising Limb	158
Figure	5.21	Flow Chart of GDANN-IV Model	159
Figure	5.22	Flow Chart for SOM(4) Model	162
Figure	5.23(a)	Observed and Predicted Flows during Testing – From GDANN-IV Model	168
Figure	5.23(b)	Observed and Predicted Flows during Testing – From GDANN-V Model	168
Figure	5.24(a)	Observed and Predicted Flows from the GDANN-IV Model –From the Year 1986	169

Figure	5.24(b)	Observed and Predicted Flows from the GDANN-IV Model – From the Year 1989	169
Figure	5.25(a)	Observed and Predicted Flows from the GDANN-V Model – From the Year 1986	170
Figure	5.25(b)	Observed and Predicted Flows from the GDANN-V Model –From the Year 1989	170
Figure	6.1	Scatter Plot: H1 Response v/s Base Flow	188
Figure	6.2	Scatter Plot: H2 Response v/s Surface Flow	188
Figure	6.3	Scatter Plot: H3 Response v/s Infiltration	189
Figure	6.4	Scatter Plot: H4 Response v/s Surface Flow	189

LIST OF TABLES

Table No.	Caption	Page No.
Table 4.1	Monthly and Daily Potential Evapotranspiration due to Hann (1972)	70
Table 4.2	Time Area Diagram Ordinates for Kentucky River Watershed	78
Table 4.3	Green - Ampt Infiltration Parameters	96
Table 4.4	Performance Statistics from CRR Models	104
Table 5.1	Performance Evaluation Indices from BANN and GANN Models	121
Table 5.2	Number of Data Points in Low, Medium, and High Magnitude Flow Categories	129
Table 5.3	Performance Statistics for Low Medium & High Magnitude Flows from BANN and GANN Models during Training Data	130
Table 5.4	Performance Statistics for Low Medium & High Magnitude Flows from BANN and GANN Models during Testing	131
Table 5.5	Details of GDANN Model Structures	160
Table 5.6	Performance Evaluation Indices from GDANN Models during Training	163
Table 5.7	Performance Evaluation Indices from GDANN Models during Testing	164
Table 6.1	Correlation Statistics of the Hidden Neurons and Observed Input Variables	182
Table 6.2	Correlation Statistics of the Hidden Neurons and Conceptual Components of the Hydrologic Process	184

LIST OF SYMBOLS

e_p	average error at the output layer
f	potential infiltration rate
i^*	winning neuron
y	output from a neuron
$AARE$	average absolute relative error
C_1, C_2	reservoir routing coefficients
E	Nash-Sutcliffe efficiency
E_{per}	coefficient of persistence
ER	effective rainfall
ER^*	updated amount of inflow coming into surface store
ET	actual daily evapotranspiration
ET_p	potential daily evapotranspiration
F	cumulative infiltration
H	hidden neuron
K	saturated hydraulic conductivity of the soil matrix
K_1	attenuation constant
Kf_t	decay ccefficient
KG_t	recession coefficient for base flow
MBE	mean bias error
$\%MF$	percentage bias in maximum flow
Net	net input received by a neuron
$NMBE$	normalized mean bias error

$NRMSE$	normalized root mean square error
P	rainfall
P_c	probability of crossover
P_m	probability of mutation
Q	predicted stremflow
QG	base flow
QO	observed streamflow
QS	surface flow
S	maximum depth of soil storage
SA	available soil moisture
S_e	effective saturation
TS	threshold statistics in percentage
α	momentum coefficient
β_0, β_1	soil moisture accounting regression equation constant coefficients
θ	soil moisture content
θ_e	effective porosity
η	porosity
η_l	learning rate coefficient
η_c	distributive index for SBX crossover operator in RGA
η_m	distributive index for mutation operator in RGA
ψ	soil suction head
ΔF	incremental infiltration
ΔFA	actual incremental infiltration amount

Chapter 1

Introduction

1.1 General

Water is essential to all kinds of lives on the earth. The total quantity of available water is estimated to be about 1386 million cubic kilometers (MKm^3). Out of all this water, only10.6 MKm^3 is available as fresh water on land, and the rest is contained either in the oceans (97%), or in the form of frozen ice on mountain tops and glaciers (Subramanya, 1994). The fresh liquid water useful for human-beings is available in rivers and reservoirs as surface water or groundwater in aquifers. The total fresh liquid water available has remained constant over the years but the needs of water are increasing by the day due to population growth, economic developments, urbanization, and other factors. If the available water resources are not utilized efficiently and effectively, the water demand will exceed the available water supply sooner rather than later. The policy makers all over the World have realized the gravity of the situation and several apex organizations like World Water Council (WWC), Global Water Partnership (GWP), and World Water Forum (WWF) have been set up with an objective to promote research and technological advancements in efficient use of the available water (Thatte, 2002).

A key component in any water resources planning, development, design, operation, or management project is the accurate estimation of the available water at a local source

such as a river. If the estimated runoff at a location in a river is inaccurate, it may lead to incorrect policy being implemented ultimately resulting not only in the loss of revenue but also in the loss of life and property in extreme cases. Runoff forecast models are useful in many water resources applications such as flood control, drought management, operation of water supply utilities, optimal reservoir operation involving multiple objectives of irrigation, hydropower generation, water supply, etc., and design of various hydraulic structures such as dams, bridges, and culverts, etc. Runoff forecasts are normally made through the development of runoff forecast models that use only hydrologic data, or through rainfall-runoff models that use both hydrologic and climatic data. An extensive evaluation of the available rainfall-runoff modeling techniques, in an attempt to achieve better model performance, is the subject of the current research effort. The thesis begins with a brief introduction to the rainfall-runoff process, techniques to model the process in the past and present, and the problems related to them.

1.2 Rainfall-Runoff Modeling

Modeling of the rainfall-runoff process has been the subject of research among hydrologists and engineers for a very long time. The transformation of rainfall into runoff is an extremely complex, dynamic, and non-linear process which is affected by many factors which are often inter-related. The factors affecting the runoff response of a watershed subjected to rainfall input include (a) storm characteristics *i.e.* intensity and duration of rainfall event, (b) watershed characteristics *i.e.* size, shape, slope, and storage characteristics of the watershed, percent of the watershed contributing runoff at the outlet at various time steps during a rainfall event, *etc.*, (c) geo-morphological characteristics of a watershed *i.e.* topography, land use patterns, vegetation, soil types,

etc., that affect infiltration, and (d) climatic characteristics such as temperature, humidity, wind characteristics, *etc.* The influence of these factors and many of their combinations in generating runoff is an extremely complex physical process and is not understood clearly (Zhang and Govindaraju, 2000). Many of the rainfall-runoff models rely on the fact that the dynamic effects of various factors mentioned above are embedded in the rainfall and runoff data. The two major steps involved in the transformation of rainfall into runoff are (i) calculation of infiltration and other losses and estimation of the effective rainfall, and (ii) subsequent transformation of the effective rainfall into runoff through an operator which simulates the behaviour of the watershed under consideration. The second step is actually responsible for modeling the non-linear, dynamic, and complex nature of the rainfall-runoff process. During this process, the input (i.e. effective rainfall) to the system (i.e. watershed) goes through two operators: (i) 'translation' in time due to variable source areas of the watershed contributing runoff at the outlet at different times, and (ii) 'attenuation' due to the storage characteristics of the watershed. All models of rainfall-runoff process, which are based on physical concepts attempt to account for these two operators with varying degree of sophistication and complexity.

The approaches used for runoff forecasting cover a wide range of methods from completely black-box models to very detailed conceptual models (Porporato and Ridolfi, 2001). Historically, hydrologists and researchers have employed two types of models: (a) deterministic models that consider the physics of the underlying process, and (b) systems theoretic/black-box models that do not consider the underlying physics of the process. Deterministic models use the equations of mass, energy, and momentum to describe the movement of water over land surface, and through the unsaturated and

saturated zones of the earth. The resulting system of partial differential equations is then solved numerically at all points in a two or a three dimensional grid representation of the watershed. Deterministic models of varying degrees of sophistication and complexity have been developed based on distributed moisture accounting for soil elements, and can be found in abundance in the literature reported in the past. Deterministic models consist of very complex structures and their development requires a lot of experience and expertise. Hydrologists have responded to these limitations by developing conceptual rainfall-runoff (CRR) models, in which, instead of using the equations of mass, energy, and momentum to describe the process of water movement, a simplified, but plausible conceptual representation of underlying physics is adopted. These representations frequently involve several inter-linked storages and simplified budgeting procedures, which ensure that at all times a complete mass balance is maintained among all inputs, outputs, and inner storage changes.

Perrin *et al.*, (2001) carried out an extensive study on many conceptual daily rainfall runoff models over a wide range of catchments throughout the world and concluded that simple models have performance levels similar to complex models with more parameters. Birikundavyi et al. (2002) recently reported the results from a CRR model called PREVIS to forecast daily flows in river Mistassibi. In this study, they reported that modifying the forecasted runoff obtained from PREVIS with the help of recently observed discharges results in better performance. It is to be noted that the deterministic/conceptual models are more suitable for modeling virgin flows, and are not able to perform adequately in watersheds altered due to manmade or other related activity (e.g. many storage structures), which essentially affect the true dynamic nature of the rainfall-runoff relationship in a watershed. Campolo et al. (1999) also concluded

that the capacity of a watershed to respond to a perturbation is more accurate when recently observed discharge values are used in the modeling process of a hydrologic system. System-theoretic models allow the use of past flows to be incorporated in the modeling procedure with ease. Moreover, many of the deterministic or conceptual rainfall-runoff models need a large amount data for calibration and validation purposes and are computationally extensive. As a result, the use of deterministic or conceptual models of the rainfall-runoff process has been viewed rather sceptically by researchers and has not become very popular (Grayson *et al.*, 1992). This has caused the attention of the hydrologists to focus on a separate category of models called systems theoretic models.

Systems theoretic models, also known as the black box models, attempt to develop relationships among input and output variables involved in a physical process without considering the underlying physical process. Systems theoretic models can be further classified into two types, (a) models with pre-defined structures but unknown parameters, and (b) models with unknown structures and unknown parameters. Conventional stochastic models (ARIMA type) come under the first category; whereas, the Artificial Neural Network (ANN) models fall under the second category. Conventional systems theoretic models (ARIMA type) suffer from being based on the linear systems theory, and may only be marginally suitable in capturing the highly complex, dynamic, and non-linear nature of the rainfall-runoff process. Recently, ANNs have been employed as alternative tools in developing non-linear systems theoretic models of the hydrological process. In recent years, the ANN technique has become increasingly popular in hydrology and water resources among researchers and practicing engineers alike. This popularity can be gauged by the plethora of studies that

have dealt with the application of ANNs in hydrology and water resources (Raman and Sunilkumar, 1995; Sajikumar and Thandaveswara, 1999; Tokar and Johnson, 1999; Tokar and Markus, 2000; Jain et al., 2001; Jain and Ormsbee, 2002; and Birikundavyi et al., 2002). Many studies have demonstrated that the ANNs are excellent tools to model the rainfall-runoff process and can perform better than the conventional modelling techniques (Smith and Eli, 1995; Minns and Hall, 1996; Shamseldin, 1997; Dawson and Wilby, 1998; Campolo *et al.*, 1999; Zhang and Govindaraju, 2000; and Jain and Indurthy, 2003). ASCE task committee (ASCE, 2000a, b) has given an excellent review of the application of ANNs to hydrology in their report.

1.3 Certain Issues in Rainfall Runoff Models

There are certain issues in the existing rainfall runoff models that are either completely ignored, or have only been partially investigated and need attention while developing rainfall-runoff model of a watershed. The motivation of the present thesis work mainly stems from these issues which are either unexplored or are partially explored. These issues are discussed in the following sections.

1.3.1 Issue of Calibration of Rainfall-Runoff Models

The parameters of a conceptual rainfall runoff (CRR) model need to be first calibrated before it can be used for predicting runoff. Calibration of CRR models has been the subject of research among hydrologists. Normally, rainfall and runoff data for representative storms taken from a watershed under consideration are employed for this purpose. Traditionally, hydrologists have employed the method of least squares or some other classical optimization method to estimate the unknown parameters of a CRR model. The classical optimization methods are deterministic in nature in searching for

the global optimal solution to an optimization problem. Recently, genetic algorithms (GAs) have been proposed as an alternative methodology for the solution of non-linear optimization problems in engineering and sciences. The GA is not purely a deterministic technique and combines random search procedures in an attempt to obtain global optimal solutions. The use of the new technique of GAs for finding the optimal parameters of a CRR model have been lacking among hydrologists barring a few exceptions.

1.3.2 Issue of Integrated (or *Grey-Box*) Rainfall-Runoff Models

Historically, researchers have employed techniques available for rainfall-runoff modeling, *viz.*, deterministic or systems theoretic, in isolation. Most of the physically based rainfall-runoff models reported in literature use only deterministic or conceptual techniques. Most of the systems theoretic models reported earlier are purely black box models in the sense that they are based in the true spirit of black box models by not considering the underlying physical process. However, it must be realized that the rainfall-runoff process is an extremely complex, dynamic, and non-linear process consisting of many components and physical variables with high degree of spatial and temporal variability, and it may be possible to achieve better model performance if the non-linear systems theoretic models are developed with some conceptual component embedded in them. Such an integrated approach will be able to take advantage of both deterministic and systems theoretic techniques. A systems theoretic model that is not purely a black box, but does incorporate the physics of the problem even in a partial sense, may be termed a *grey-box* model as some of the details are visible in the form of the involvement of physics, though in a partial sense. Such type of models can also be termed as integrated models as they take advantage of, and integrate, the two techniques.

In the context of rainfall-runoff modeling, it may be possible to elevate a black box model to a *grey-box* model by physically modeling some of the components of the rainfall-runoff process such as evapotranspiration, interception, infiltration, and snowmelt, etc., and embed such component(s) into a non-linear systems theoretic rainfall-runoff model. For example, consider the following: Most of the non-linear systems theoretic models for rainfall-runoff process have employed total rainfall as one of the input variables. However, it must be noted that for the same value of total rainfall one may get a very wide variation in effective rainfall values, depending on the antecedent moisture conditions (Kumar and Minocha, 2001). A *grey-box* model for rainfall-runoff process can be developed by modeling the infiltration process using conceptual infiltration equation, which accounts for the varying soil moisture conditions (such as Green-Ampt equations), computing the effective rainfall at each time step, and then including the effective rainfall in the input vector in the ANN model instead of total rainfall. Such an approach not only forms a strong basis of including physics in the systems theoretic models but also minimizes the possibility of any errors in the runoff forecasts due to varying antecedent moisture and initial conditions of the watershed. Embedding one or more conceptual components of the physical process into the non-linear systems theoretic model essentially helps the model in understanding the complex nature of the non-linear and dynamic physical process in a better way as part of the overall job is already performed using physically based techniques. Therefore, including effective rainfall as an input variable in an ANN rainfall runoff model may help in capturing the dynamic nature of the complex rainfall-runoff process in a better way as compared to the case when using total rainfall but it needs to be explored.

1.3.3 Issue of Modeling Decomposed Flow Hydrograph

An important aspect that affects the performance of the rainfall-runoff models using ANNs to be employed in operational hydrology is applying a single method or technique for modeling different components of a physical process having different dynamics. Most of the ANN applications reported in literature attempt to model the complex, dynamic, and non-linear rainfall-runoff process represented in a flow hydrograph, using a single ANN. However, it must be realized that the runoff response of a watershed, represented by different segments of a flow hydrograph, is produced by different physical processes ongoing in a watershed. For example, the rising limb of a flow hydrograph is the result of the gradual release of water from various storage elements of a watershed (e.g. surface and sub-surface storage) due to gradual repletion of the storages due to the rainfall input. The rising limb of the hydrograph is influenced by varying infiltration capacities, watershed storage characteristics, and the nature of the input i.e. intensity and duration of the rainfall, and not so much by the climatic factors such as temperature and evapotranspiration etc (Zhang and Govindaraju, 2000). On the other hand, the falling limb of a hydrograph is the result of the gradual release of water from the watershed after the rainfall input has stopped, and is influenced more by the storage characteristics of the watershed and climatic characteristics to some extent. Therefore, the use of a single ANN to represent the input output mapping of the whole hydrograph may not be as efficient and effective as compared to developing two different mappings representing the two limbs of the hydrograph. Further, it has also been reported in the literature that rainfall-runoff mapping in a watershed can be fragmented or discontinuous having significant variations over the input space because of the functional relationships between rainfall and runoff being quite different for low, medium, and high magnitudes of streamflow (Zhang and Govindaraju, 2000). In order

to capture such fragmented and/or discontinuous relationships, they proposed a modular neural network by decomposing the complex rainfall-runoff mapping problem into several simple problems, each of which can be solved using a simple ANN. They found the performance of the developed modular neural network to be better when compared to a fully connected feed-forward network. In a study, Furundzic (1998) used self-organizing map (SOM) classifier to decompose the input space into three classes corresponding to different dynamics of the rainfall-runoff process. Apart from these two studies, the efforts in the area of using decomposition techniques to decompose the input output data space and develop models for different segments of the rainfall-runoff process have been lacking. Moreover hardly any study that explores the ability of ANNs in capturing different functional relationships in different segments of a flow hydrograph dominated by different dynamics of the physical processes.

1.3.4 Issue of Training of ANN Rainfall-Runoff Models

Most of the ANN applications reported for rainfall-runoff modeling have employed the back-propagation (BP) training algorithm proposed by Rumelhart et al. (1986) and its variations. Raman and Sunilkumar (1995) trained ANNs by employing momentum correction factor; Shamseldin (1997), and Thirumaliah and Deo (2000) implemented conjugate gradient algorithm; Hsu et al. (1995) incorporated linear least square simplex method to enhance the speed of training and trying to ensure near global solutions. Some other examples of using BP training algorithm for rainfall-runoff modeling include Lorrai et al. (1995), Campolo et al. (1999), Rajurkar et al. (2002), Birikudavyi et al. (2002), and Jain and Indurthy (2003). The use of gradient search techniques, such as those employed in these studies, often result in inconsistent and unpredictable performance of the neural networks (Curry and Morgan, 1997). Hsu et *al.* (1995)

developed rainfall-runoff models using ANNs, and experienced that the ANN models trained using BP algorithm under-predicted low flows and over-predicted medium flows. While discussing their results, Hsu et al. (1995) mentioned that this may have been due to the models not being able to capture the non-linearity in the rainfall-runoff process and suggested that there is still room for improvement in applying different algorithms to reach near global solutions, and achieve better model performances. Ooyen and Nichhuis (1992) tried to improve the convergence during training of the ANNs using BP algorithm and experienced that the convergence was slow and the learning process was inefficient when the output data contained values near to zero or unity. Sajikumar and Thandaveswara (1999), and Tokar and Markus (2000) also experienced that the patterns with target values in the neighborhood of zero will not learned properly by BP algorithm in rainfall-runoff modeling. Clearly, there seems to be a strong need to explore alternative training methods to train an ANN to develop better models for rainfall-runoff process for operational purposes. The training of an ANN is primarily a non-linear optimization problem in which the objective is to minimize the global error at the output layer. The method of training of an ANN needs to be very robust because of one or more of the following difficulties during the search for the global optima: (a) there may be several major regions of attraction into which a search strategy may converge, (b) each of these major regions of attraction may contain many local optima that may be either close to or far away from the global solution, (c) the error function surface may not be smooth but vary in an unpredictable manner, and (d) the weight elements of the weight matrix may exhibit varying degree of sensitivity and large non-linear inter-dependence. Any non-linear optimization problem with the above characteristics must be solved with a global optimization strategy that is based on the following concepts: (a) combination of deterministic and probabilistic approaches, (b)

systematic evolution of the solution in the direction of the global improvement, and the concept of competitive evolution (Duan et al., 1993).

In recent years, a variety of evolutionary computing techniques have been proposed to solve the problems in common engineering applications. One of the approaches in the evolutionary computation used in the field of engineering is genetic algorithm (GA). The GAs are not problem specific and do not require continuous differentiable functions for optimizing the objective functions, as required in the BP training method. There are many studies involving the application of GAs in hydrology and water resources but there is not a single study exploring the use of GAs to train ANNs in hydrology and water resources. Moreover, most of the GA applications in engineering and hydrology use binary coded GA. The binary coded GA suffers from certain shortcomings such as having to work with discrete search spaces for problems involving continuous search spaces, low precision, hill climbing, hamming cliffs, positional bias, and no control on infeasible solutions being created while moving from one generation to the next *etc.* (see Chapter 3 for details). In order to overcome such problems of binary coded GA, *real-coded* GA can be employed (Deb and Agarwal, 1995; and Deb, 2001). It is to be noted that a strong need to experiment with the *real-coded* genes has been emphasized in common engineering applications involving continuous physical variables because the *real-coded* GA not only overcomes the problems associated in the binary-coded GA but also allows the use of special genetic operators developed for them (Michalewicz, 1996). Zhang and Govindaraju (2000) have pointed out that the rainfall-runoff mapping in a watershed can be fragmented or discontinuous having significant variations over the input space because of the functional relationships between rainfall and runoff being quite different for low, medium, and high magnitudes of streamflow. They found that

the performance of a single ANN rainfall-runoff model trained using BP method was worse than the modular neural network. However, it must be realized that it may be possible to capture the complex fragmented input-output mapping, such as that for the low, medium, and high flows, using a simple feed-forward network trained using a more rigorous and aggressive search technique of *real-coded* GA that is based on the combination of deterministic and probabilistic approaches, and a systematic evolution of the search in the direction of the global improvement. However, it needs to be investigated in the context of developing rainfall-runoff models for operational hydrology.

1.3.5 Issue of Modeling Low Magnitude Flows

Modeling of low magnitude flows has been experienced by the researchers to be the most problematic, as pointed out in the previous section. The problems in modeling the low magnitude flows cause the overall rainfall-runoff relationship to be not very efficient and accurate. When such poor rainfall-runoff relationships are used in predicting runoff values, they tend to be either under-predicted or over-predicted due to poor generalization. The fact that a major fraction of the total runoff record for most watersheds consists of low magnitude flows further aggravates the problem of poor generalization when low magnitude flows are not modeled properly. This may be because of the fact that the dynamics of the physical processes inherent in the low magnitude flows are more complex than that for the high magnitude flows. For example, during low magnitude flows, the watershed and moisture conditions are close to the drier side and the rainfall-runoff relationships are more dominated by the varying infiltration capacities, land use and soil conditions, and the varying storage characteristics of the watershed. Further, when the moisture and watershed conditions

are on the drier side then the travel times are on the higher side making the overall rainfall-relationship to be more dynamic and complex. On the other hand, when the moisture and watershed conditions are close to saturation during the high magnitude flows, then the infiltration plays minimal role and most of the rainfall becomes effective rainfall, which quickly travels towards the outlet to become runoff. Therefore, modeling of the low magnitude flows needs extra care especially when using the non-linear systems theoretic technique of ANNs. It may be possible to model the low magnitude flows more accurately by using a different method for training the ANN rainfall-runoff models, or by using integration of deterministic and non-linear systems theoretic techniques while modeling the complex, dynamic, and non-linear rainfall-runoff process in order to achieve better model performance; however, it needs to be investigated.

1.3.6 Issue of Physics inside a Trained ANN Model

Despite their numerous advantages like universal function approximation property, robustness, and ability to learn, ANNs receive major criticism due to their weakness of being black-box models. This criticism mainly stems from the fact that no satisfactory explanation of their internal behavior has been offered yet. This is a significant weakness, for without the ability to produce comprehensive decisions it is hard to trust the reliability of networks addressing real-world problems (Benitez, et al., 1997). However, it must be realized that if the non-linear function being mapped by the ANN is explored further, one may be able to shed some light into the physical processes of the system being modeled that is inherent in a trained ANN. It is quite possible that while training of an ANN, different hidden neurons do learn the individual relationships inherent in different portions of a flow hydrograph. However, identifying components

of a physical process in a trained ANN is an area of research that is virgin and remains unexplored, especially in hydrology. There are many techniques, e.g. sensitivity analysis and correlation analysis, which can be employed to analyze a trained ANN to explore for the possibilities of physics embedded inside an ANN. For example, the responses from individual hidden neurons in a trained ANN may be correlated with the conceptual components of the hydrologic process along with the input vector. Therefore, it may be possible to elucidate the physical process being mapped by an ANN by examining the knowledge about a given problem domain in the form of different conecptual components of the physical process along with the input variables, in relation to the distributed components of the ANN. However, this needs to be explored in the context of ANN rainfall runoff models.

1.4 Objectives of the Thesis

This thesis is an attempt to extensively evaluate various techniques available for rainfall-runoff modeling, and develop improved methodologies for capturing the complex, dynamic, non-linear, and fragmented nature of the rainfall-runoff process. Rainfall-runoff models using a wide variety of techniques ranging from simple conceptual/deterministic technique, to non-linear systems theoretic technique of ANNs, *real-coded* genetic algorithm, and their combinations will be investigated. The issues related to rainfall-runoff modeling in a watershed discussed above form the core of the objectives of the present thesis work. The primary objectives of the present thesis are summarized below.

1) Develop conceptual rainfall runoff models that are capable of capturing the complex, dynamic, and non-linear rainfall-runoff process in a large watershed.

2) Explore the use of *real-coded* genetic algorithm to calibrate parameters of the conceptual rainfall-runoff models.
3) Develop ANN models of the rainfall-runoff process of the black box type, and compare their performances with the conceptual rainfall runoff models.
4) Explore the possibility of developing *grey-box* ANN rainfall-runoff models with conceptual component(s) embedded in them for use in operational hydrology, and compare their performances with the ANN rainfall-runoff models of black box type.
5) Explore the use of *real-coded* genetic algorithm to train ANN rainfall-runoff models of both black box and *grey-box* types for use in operational hydrology.
6) Investigate the validity of the hypothesis that the use of either soft or physics based heuristic decomposition of the input output data, and the development of rainfall-runoff models for each decomposed class corresponds to different dynamics of the rainfall-runoff process, and therefore, results in better overall model performance.
7) Investigate for the presence of physics inside a trained ANN rainfall-runoff model by exploring the parallel distributed components of the ANN model in relation to the conceptual components of the hydrologic process.

A secondary objective of the thesis is to validate the hypothesis that ANN rainfall-runoff models are not simply black boxes in an effort to establish them as the powerful tools for modeling the complex, dynamic, non-linear, and fragmented rainfall-runoff process when used properly in conjunction with conceptual and other soft computing techniques. Another secondary objective of the thesis is to initiate an extensive programme of research that will focus on the use of the relatively new techniques of

ANNs and the *real-coded* GAs for the betterment of rainfall-runoff process modeling. This would help in producing more accurate runoff forecasts that can be employed in various water resources planning, development, design, operation, and management projects. The daily total rainfall (mm) and daily mean streamflow (m^3/s) data derived from the Kentucky River watershed in Kentucky, USA will be employed to develop and test the models and methodologies proposed in this thesis.

1.5 Organization of the Thesis

The first chapter presents the introduction of the overall problem of the complex, dynamic, and non-linear rainfall-runoff process, techniques available to model the process, certain issues associated with the existing rainfall-runoff models, and the objectives of the present research work. Chapter 2 presents the literature review of the rainfall-runoff modeling and closely related areas. Chapter 3 presents a brief introduction to the relatively new techniques of ANNs and the GAs including *real-coded* GA that will be employed in developing various rainfall-runoff models in the present research work. Chapter 4 and Chapter 5 discuss in details, the principles and procedures followed in the development of CRR models and ANN rainfall-runoff models, respectively; and the results obtained. Chapter 6 presents the procedures and findings of the preliminary study on the exploration of physical significance in the ANN rainfall-runoff models. The final chapter, Chapter 7, presents the summary of the work carried out, the conclusions of the thesis work, the limitations, and scope for the future work. References and appendices are provided at the end.

Chapter 2

Review of Literature

2.1 General

Modeling of hydrological process has been the subject of interest among researchers for a very long time. The involvement of many physiographic and climatic factors makes the hydrological process more complex to understand, especially the rainfall-runoff process. Considerable research has been carried out in developing mathematical models of the rainfall-runoff process in the past. Historically, the kind and variety of mathematical models for rainfall-runoff process have ranged from simple conceptual models to more distributed deterministic models that consider the physical processes involved in the hydrological phenomena. System theoretic models have also been developed, which do not consider the underlying physical processes. Stochastic models of ARIMA type and Artificial Neural Network (ANN) models fall under the systems theoretic category of rainfall-runoff models. ANNs have been proposed as efficient tools for modeling and forecasting in recent years. This chapter reviews the previous work on conceptual and ANN modeling aspects of rainfall-runoff process and also an attempt is made to identify the shortcomings in the present state of knowledge in the area of hydrological modeling. The first section of this chapter deals with the conceptual models, while the second section deals with ANN rainfall-runoff models reported in literature.

2.2 Conceptual Rainfall-Runoff Models

The literature review of the conceptual rainfall-runoff (CRR) models is divided into two sections. The first section deals with the development and application of the CRR models; whereas, the second section describes the studies related to the calibration of CRR models.

2.2.1 Development and Application of CRR Models

The origin of conceptual modeling of the rainfall-runoff process started from the event based rainfall-runoff model developed by Mulvany (1850) called Rational Method. Later, Sherman (1932) proposed the concept of Unit hydrograph. He assumed that the runoff process was linear and time invariant. The unit hydrograph and its subsequent evolution to the instantaneous unit hydrograph provided a basis for the storm response models of Nash (1957) and Dooge (1959). After that a large number of event based streamflow simulation models were developed such as The Illinois Urban Drainage Area Simulator (ILLUDAS, Terstriep and Stall, 1974), PSRM (Lakatos, 1976), RAFTS (Goyen, 1987). The most comprehensive and widely used event based rainfall-runoff model is the HEC-1 flood hydrograph model, developed by Hydrologic Engineering Center (1990). This computer package can be used to simulate a direct runoff hydrograph from a watershed by representing the watershed with interconnected hydrologic and hydraulic components. In HEC-1, many elements of the hydrological process can be modeled using several options. Infiltration can be estimated using five options: (a) initial and uniform loss rate, (b) exponential loss rate, (c) SCS curve number method (d) Holtan's infiltration equation, and (e) Green Ampt method. The direct runoff hydrograph can be estimated using the unit hydrograph method and the kinematic wave method. A univariate search technique is employed to determine an

optimal set of model parameters. Normally, event based rainfall-runoff models provide information on the peak and time to peak etc., which are more useful from engineering design point of view, e.g. to design bridges, culverts, and storm water drains etc. Continuous streamflow prediction is needed for operation and management purposes.

In the literature, many continuous CRR models have been proposed that are based on either distributed approach or a lumped approach. In a lumped approach, the spatial variations in various influencing hydrologic and climatic variables are ignored and the various watershed conditions are assumed to be homogeneous and uniform. In contrast, the distributed approach considers the dispersed information of catchement characteristics. Some notable examples of physically based distributed models from the computer age include the System Hydrologique European (SHE) model (Abbot et al., 1986), the Institute of Hydrology Distributed Model (IHDM), (Beven et al., 1987), and SWATCH model (Morel-Seytoux and Al Hassoun, 1989). The distributed rainfall-runoff models consist of many parameters and require a lot of physical data of the watershed at the grid scale. Although this approach might be useful in terms of knowledge of the processes, it has limitations when applied in an operational context due to its complexity. Because of the limitations of distributed models and the search for more practical deterministic rainfall-runoff models have resulted in framing lumped approach, know as "conceptual models".

Conceptual models can predict daily, monthly, or seasonal estimates of streamflow on a continuous basis. In a conceptual model the transformation of input into output is determined by assuming that the whole system consists of several interconnected subsystems, each representing a certain component of the hydrologic process.

Empirically or heuristically determined, but physically realistic functions are used to describe the internal operation of the process. Stanford Watershed model (SWM) was probably the first comprehensive conceptual watershed simulation model developed by Crawford and Linsley (1960) capable of producing streamflow forecasts on a continuous basis. This is a deterministic model that uses precipitation and potential evapotranspiration as meteorological inputs and produces time variant predictions of streamflow and ground-water storage as outputs. The Storm Water Management Model (SWMM) was the first comprehensive computer model (Metcalf and Eddy, 1971) for analysis of quantity and quality problems. SWMM is capable of simulating both event and continuous based hydrologic processes. With this beginning number of conceptual models were developed for continuous simulation of the rainfall-runoff process. Some of the CRR models developed from last two decades with varying degree of sophistication and complexity include SCM (Refsgard, 1981), PDM (Moore and Clarke, 1981), IHACRES (Jakeman et al., 1990), MODHYDROLOG (Chiew and McMahon, 1994), HBV (Bergström, 1995), TANK (Sugawra, 1995), TOPMODEL (Beven et al, 1995), Xinanjang (Zhao and Liu, 1995), SMAR (Tan and O'Connor, 1996), mSFB (Summer et al., 1997), Arno (Todini, 1996), IHACRES (Littlewood et al., 1997), the SWAT model (Arnold et al., 1998), and GR3J (Edjanto et al., 1999).

Perrin *et al.*, (2001) made a comparative assessment of nineteen different lumped structures of the daily CRR models with varying number of parameters, 9 from the MODHYDROLOG model (Chiew and McMahon, 1994) to 3 parameters from GR3J (Edijantno et al., 1999). Their study employed rainfall and runoff data from 429 different watersheds taken from all over the world. They examined the role of complexity in hydrological models by studying the relation between the number of

optimization parameters and model performance. The authors demonstrated that simple models have the performance levels similar to those from the complex models with more parameters. They suggested that up to 3 to 5 free model parameters are sufficient to model the complex rainfall-runoff process. Moreover, they emphasized that the principle of parsimony should not be enforced at the cost of accuracy of the model in prediction. The authors also indicated that there cannot be a model that could be the best under all the circumstances. More recently, some researchers have modified the output of conceptual model with the previously observed discharges in order to account for the continuously changing dynamics of the watershed. Birikundavyi et al. (2002) demonstrated that the forecasts from a conceptual model called PREVIS, when modified using the recently observed streamflows, result in a better performance.

The experiences of Perrin et al (2001) and Birikundavyi et al. (2002) that (a) simple CRR models with fewer parameters are equally good when compared to more complex mathematical models with many parameters, and (b) the changing dynamics of the watershed, represented in the past observed flows, need to be incorporated in the modeling of the non-linear, complex, and dynamic rainfall-runoff process. It is possible that, both of these objectives can be achieved by developing a mathematical framework to represent the hydrological process in which the hydrological process or its components are modeled using conceptual techniques and the historically observed flows are employed as input in the overall rainfall-runoff modeling framework. Such an approach will be better as compared to the one that was employed by Birikundavyi et al. (2002), where the streamflow forecasts are modified using linear proportions of past times, and the non-linear dynamics of past rainfall and flows on the future flows are ignored. Regardless of the kind of mathematical framework for representing the

rainfall-runoff process in a watershed, the CRR model needs to be calibrated first before it can be applied for forecasting purposes. The review of literature on the calibration of CRR models is presented in the next section.

2.2.2 Calibration of CRR Models

The successful application of CRR models depends upon how well the model is calibrated. The calibration of CRR models has been researched extensively over the last four decades. In the early stages of conceptual modeling, researchers relied on trial and error methods aided by observation and experience, to obtain the parameters that represent the best fit among the observed and estimated data. Historically, modelers have used least squares principle to determine parameters of a CRR model. Dawdy and O'Donnell (1965) used Rosenbrock (1960) optimization algorithm and developed computer programs to optimize parameters of CRR models automatically. Consequently, with the availability of high spped computers, sophisticated techniques became popular. Most of the CRR models from computer age have automated calibration capability embedded in them. However, automated approaches for calibration have received much attention in the last decade. The determination of the parameters of a CRR model is essentially an optimization problem in which errors between observed and computed outputs are minimized subject to the non-negativity and other constraints ensuring that the equations representing the transformation of rainfall into runoff are incorporated in the overall optimization program. Many of the components of the rainfall-runoff process are non-linear in nature and the optimization program may involve non-linear equality constraints. The resulting non-linear optimization problem thus normally consists of complex non-convex search space with many local minima. This causes difficulties in the application of automated approaches.

The difficulty in application of automated calibration to some of the models is reported by; Pickup (1977), Sorooshian and Gupta (1983), Sorooshian and Gupta (1985), and Henderickson *et al.* (1988). Automatic calibration procedures in earlier usage are not capable of finding optimal parameter estimates with confidence, leading to inaccuracy in model forecasts. The theory and practice of global optimization progressed rapidly and presently a wide variety of different algorithms are available, because of the improved and affordable technology of computational capability.

Masri *et al.*,(1978), Pronzato *et al.*, (1984) and Brazil and Krajewski (1987), used uniform random search, and the adaptive random search (ARS) methods for fine tuning of the parameters of the SAC-SMA model and concluded that ARS method was an alternative tool to nonrandom search techniques. Duan *et al.*, (1992) tested the performance of combined adaptive random search (ARS) and simplex method, Multistart Simplex Method and Shuffled Complex Evolution (SCE-UA) method on the SIXPAR model, which is simplified version of the SAC-SMA model. The authors pointed out that SCA-UA method is more consistent in locating the global optima of the SIXPAR model. Further, Duan *et al.*, (1993) tested the algorithm on theoretical functions and SIXPAR CRR model, and concluded that the SCE algorithm is more effective and efficient in locating global optimum for a broad class of problems. Yapo et al. (1995), studied sensitivity of the data period for calibrating CRR models using SCE-UA global optimization method and concluded that at least 8 years of data are required for calibration to make the model parameters insensitive to calibration data period. Gan and Biftu (1996) tested SCE-UA, and the Multiple Start Simplex (MSX), and the local Simplex methods by calibrating four different CRR models on eight catchments and

found that the SCE-UA and local Simplex methods are viable optimization tools and MSX is inefficient computationally.

There is always a possibility of multiple solutions, which is a strong feature of non-linear problems. In such an event, the set of parameters that yield the lowest value for the objective will be the optimum solution. Such a solution is obtained by scanning the acceptable solutions rather than through application of any rigorous conditions. The application of Genetic Algorithm (GA) helps in scanning acceptable solutions with the help of population approach.

Wang (1991) used binary coded GA for a theoretical function optimization and to determine optimal set of parameters of Xinanjang CRR model in Bird Creek catchmnet. The author showed that the GA with further tuning by the sequential simplex method provides efficient and robust means of optimizing model parameters. In a further study, Franchini (1996) modified the GA algorithm developed by Wang (1991) by coupling it with the Sequential Quadratic Programming (SQP) in order to improve its efficiency. The author tested and proved that the GA-SQP algorithm performed better than the GA algorithm developed by Wang (1991). The author also applied GA-SQP algorithm on theoretical and practical case of calibration of "A Distributed Model" (ADM) (combination of ARNO model and XINANJIANG model) and obtained 100% success rate in finding the global optimum for the theoretical case. Moreover, in practical cases, the GA with the application of SQP produced significant reduction in the objective functional value. The author hinted that in the event that high precision is required in the definition of global minimum of objective function, the GA must be coupled to a local optimization procedure. Wang (1997) applied GA for parameter optimization in

four watersheds and emphasized that GAs are more useful search techniques, and are capable of finding the objective function value very close to the global minimum. In another study, Franchini and Galeati (1997) compared several binary coded GA schemes in comparison with Pattern Search (PS) method for calibration of ADM for a theoretical case and two cases of real data. The authors pointed out that in calibrating model parameters in a theoretical case, GA was superior compared to PS method. Further, in real world cases, the robustness of GA varied from case to case, and the authors reasoned that irrespective of its formulation, GA is comparable to but does not supplement the traditional methods. The binary representation of the variables and parameters in GA leads to less precision, resulting into many other problems (Deb and Agarwal 1995). This may be one of the reasons for only comparable performance of GA in the calibration of the ADM model.

Later, Ndiritu and Daniell (2001), improved binary-coded GA with three strategies of fine tuning, a hill climbing, and independent subpopulation searches coupled with shuffling in addition of variable fitness scaling. They applied Improved GA (IGA) to the optimization of Hartman and Griewank functions, and to calibrate SIXPAR CRR model parameters. They noticed that the improved GA performed better than the standard GA as far as locating all global optimum was concerned in all theoretical functions considered. But it was less efficient when compared to SCE algorithm in Griewank function optimization and SIXPAR conceptual model calibration. They concluded that the modified GA can be considered effective tool and should be considered on a case-to-case basis. In the cases where high precision is required, the binary representation of the variables might hinder the performance of the algorithm.

Real-coded GA can be applied to overcome many of the problems associated with the binary-coded GA.

2.3 ANNs in Hydrological Modeling

The literature review of the application of Artificial Neural Networks (ANNs) in the hydrological process modeling has been divided into two parts. The first part describes the development and application of ANNs to hydrological modeling, and the second part discusses the problems associated with the algorithms applied for training the ANNs for hydrological modeling.

2.3.1 Development and Application of ANN Models

Interest in the application of ANNs in civil engineering started in the late 1980s (Flood and Kartam, 1994 a, b). Due to their high degree of fault tolerance or robustness, and their ability to adapt and continuous learning as integrated components, ANNs have found many applications in the field of hydrology. The strengths of ANNs are well suited to hydrologic modeling, where training data sets are normally limited and noisy, and the inherent physical processes are highly complex, non-linear and dynamic. The application of ANNs to the field of rainfall-runoff modeling started in the early 1990s. Although the present study deals with daily rainfall-runoff modeling, literature review looks into other time horizons as well to highlight the essence and limitations of ANNs to rainfall-runoff modeling, in general.

The earlier studies of application of ANNs in hydrology used synthetic data. Karunanithi et al. (1994) used synthetic data for the development of ANN model for daily flow prediction. However, they used practical catchment data in some of the

validation set. Smith and Eli (1995) and Minns and Hall (1996) used synthetic domain and data for predicting hourly flows. They made a feasibility study on the capability of ANNs in prediction of runoff. The only exception was Crespo and Mora (1993), who modeled the flow with ANN using data taken from a watershed. But they did not use independent data set for training and validation. Consequently, Sajijumar and Thandaveswara (1999), Ehran *et al.*, (2000), Birikundavyi *et al.*, (2002), Jain and Indurthy (2003) developed ANN rainfall-runoff models using practical data taken from real watersheds and demonstrated that ANNs can be powerful tools for rainfall-runoff modeling. Ehran *et al.*, (2000) reported that if controlled headwaters were present in the watershed, the ANN models would be able to detect relational patterns. In such cases an important step to achieve good performance is the systematic building of the models, which was probably ignored in earlier studies. Cheng and Titterington (1994) indicated that the consideration of statistical principles in the ANN model building process might improve the performance. Another important aspect in ANN model development is the relationship between the number of free parameters and the data patterns considered which if ignored, may lead to either over-training or under-training of the network. This aspect was studied by Amari et al. (1997). They suggested that over-fitting does not occur if the number of training samples is at least 30 times the number of free parameters. Many reviews have been carried out on the feasibility and applicability of ANNs to water resources, hydrological modeling, and flood forecasting. ASCE task committee on applications of ANNs in hydrology (ASCE, 2000 a, b) discussed the preliminary concepts about ANNs and the plethora of applications of ANNs in the field of hydrology.

There are many practical situations, where the main concern is merely making accurate stream flow predictions at specific watershed locations. Many studies have proved the superiority of ANN rainfall-runoff models over other systems theoretic models (Raman and Sunilkumar 1995; Hsu et al., 1995; Lorrai et al. (1995); Sajikumar and Thandaveswara, 1999; Shamseldin 1997; Tokar and Johnson 1999; Ehran et al., 2000; Elshorbagy et al., 2000; Tirumaliah and Deo 2000; Rajkumar et al., 2002; and Jain and Indurthy 2003). Tibshirani (1994) recommended that it is often better to get an approximate solution to a real problem than an exact solution to an over simplified one. Further, ANN rainfall-runoff model performance has been found to be superior to that of conceptual models (Hsu et al. 1995; Tokar and Johnson 1999; Tokar and Markus 2000; and Birikundavyi et al., 2002). More recently, Dawson and Wilby (2001) attempted to provide transparency to the ANN models on hydrological modeling. Maier and Dandy (2000) stressed on some key issues that should be taken into consideration while applying the ANNs to hydrological modeling, and, they recommended that there is a need of systematic approach in the model development, such as selecting network, finding suitable input and output variables of the corresponding network, methodology to implement for parameter optimization, performance criteria, and model validation. Such systematic approach would result in more reliable and accurate streamflow predictions to be employed in many water resources management applications.

Identification of Input vector

In most of the studies, the input variables were determined using trial and error methods (Hsu et al., 1995; Lorrai an Sechi, 1995, Tirumalaiah and Deo, 1998), sensitivity analysis using neural networks (Furundzic, 1998), and cross-correlation analysis (Dawson and Wilby1998; and Campolo et al. 1999). It has been pointed out that defining an input vector to the network is not a trivial task and the efficiency of the

developed ANN model depends on the identification of the most powerful input output vectors. Dawson and Wilby (2001) strongly recommended the establishment of the optimal lag-interval between the process and the response. The choice of input variables is generally based on *a-priori* knowledge of causal variables in conjunction with inspection of time series plots of potential inputs and outputs (Maier and Dandy, 2000). Sudheer *et al.*, (2002) attempted to channelize the input vector identification process to ANNs using cross correlation analysis of the rainfall data and auto-, and partial auto-correlation analysis of the runoff data.

The ASCE task committee report (ASCE 2000 b) recommended to build the ANN rainfall-runoff models by considering different physical and persistence conditions of the catchment, and causative variables. Arun kumar and Minocha (2001), in their discussion, mentioned that including the effective rainfall as an input variable in the ANN models may help in capturing the dynamic nature of the complex rainfall-runoff process in a better way as compared to the case when using total rainfall. The study carried out by Rajurkar et al. (2002) considered a transformed input vector with respect to spatial variations in order to predict daily runoff. They found that embedding the transformed component in the input vector improved the prediction accuracy of the ANN model.

It must be realized that many of the causal variables affecting runoff transformation are uncertain in nature and involve large temporal and spatial variations. MacGregor (2001) suggested that the decomposition technique could be helpful in situations involving uncertainties due to complexity of the physical process and spatial & temporal variation. He suggested to decompose a complex problem into many smaller simpler problems.

Furundzic (1998) used soft decomposition of the input space of rainfall-runoff model by self-organizing map (SOM), and reported that it resulted in input number reduction of the network thereby decreasing the complexity of the ANN structure. More recently, Zhang and Govindaraju (2000) used another type of decomposition of input space by designing different modules of networks for different magnitudes of flow. They found the performance of the developed modular neural network to be better when compared to a single fully connected feed-forward network.

There is a strong belief in the scientific community that the ANN models are black-box type of models, as they do not consider the underlying physical process. Maier and Dandy (2000), and Dawson and Wilby (2001) have stressed that future direction of research on the ANN models should be on the extraction of the hydrological knowledge from the parallel distributive architecture and connection weights of trained ANN models. Dawson and Wilby (2001) also indicated that the direction of research should be to provide insights into previously unrecognized relationships within hydrological 'black boxes'. Abarahart et al. (2001) disaggregated neural network solution (saliency analysis) in terms of its forecasting inputs. The authors pointed out that the 'saliency analysis' offers new opportunities to design and test models and to gain insights into the behavior of a black box neural network approach.

2.3.2 Training Algorithms for ANN Models

An important step in the development of an ANN model is the training of the ANN using known examples or data. There are many methods available for the training of an ANN model such as Back Propagation (BP) (Rumerlhart et al., 1986 a, b), BP with momentum term (Tollenaere, 1990), second order strategies (Battiti 1992), conjugate

gradient strategies (Fitch et al. 1991; Charalambous 1992), and Cascade Correlation Algorithm (Fahlman and Lebiere 1990). Maier and Dandy (2000), ASCE task committee (ASCE 2000, b), and Dawson and Wilby (2001) have pointed out that the majority of applications in hydrology employed logistic sigmoid function, a three layer feed forward network, and back propagation training algorithm and its variations to train the network. For example Karunanidhi et al (1994), Lorrai et al. (1995), Minns and Hall (1996), Shamseldin (1997), Dawson and Wilby (1998), Campolo et al. (1999), Sajikumar and Thandaveswara (1999); Tokar and Johnson (1999), Tokar and Markus (2000), Rajurkar et al. (2002), Birikudavyi et al. (2002), and Jain and Indurthy (2003) used three layer network with BP algorithm to model the complex rainfall-runoff process. The BP method seems to be adequate for majority of the ANN applications mentioned above. However, the BP algorithm does suffer from certain limitations that have been reported in literature. Hsu et al. (1995) found the training of an ANN using BP to be very slow and incorporated linear least square simplex algorithm (LLSSIM) to enhance the speed of training, and trying to ensure near global solution. Karunanithi et al. (1994), Raman and Sunilkumar (1995), Shamseldin (1997), and Thirumaliah and Deo (2000) tried variations of the BP algorithm such as quick prop, momentum correction factor, conjugate gradient algorithm, and cascade correlation algorithm in order to achieve speed and accuracy. According to Rumelhart et al. (1986 a), the gradient search method employed in BP does not guarantee the global optimal solution. There are many situations like flat error surfaces, non-circular error contours, which can be experienced in the calibration of neural networks that hinder the speed of the training algorithm (Hassoun, 1995). Veiela and Reifman (1997) found that sigmoid-type transfer functions have the problem of premature convergence to flat slopes. Curry and Morgan

(1997) hinted that the use of gradient search techniques often results in inconsistent and unpredictable performance of the neural networks.

Campolo et al. (1999) found the training with BP algorithm difficult because of falling limb of hydrograph had more number of zeros in the rainfall. They recommended the use of recent water level information in the input vector so as to get more accurate response to a rainfall perturbation in the basin. Ooyen and Nichhuis (1992) tried to improve the convergence during training of the ANNs using BP algorithm and experienced that the convergence was slow and the learning process was inefficient when the output data contained values near to zero or unity. Hsu et al. (1995) developed rainfall-runoff models using ANNs and experienced that the ANN models under-predicted low flows and over-predicted medium flows. Sajikumar and Thandaveswara (1999), and Tokar and Markus (2000) also experienced that the patterns with target values in the neighborhood of zero did not learnt properly by the ANN using BP algorithm. All of these studies indicate that the popular BP training method is able to achieve reasonable model performance but is not capable of finding a generalized global solution which can represent the rainfall-runoff relationships inherent in low, medium, and high magnitude flows. The low magnitude flows in particular have been found to be problematic while developing ANN models of the hydrologic process. Clearly, there seems to be a strong need to explore other training methods to train ANN rainfall-runoff models that are not based on the gradient search techniques. The training of an ANN is essentially a non-linear optimization problem that may contain one or more of the following difficulties: (a) there may be several major regions of attraction, (b) each major region may contain numerous local minima, (c) the objective function surface in the multi-parameter space may not be smooth and continuous, (d) there may

be highly non-linear interdependence and mutual compensation among weight parameters, and (e) the response surface near the solution is often non-convex. Hsu et al. (1995) recommended the use of different global search techniques that combine the deterministic & probabilistic approaches such as genetic algorithms (GAs) to train the ANNs to get near global solutions.

Bäck *et al.* (1997) commented that simulated evolutions is quickly becoming the method of choice for complex problem solving especially when more traditional methods cannot be efficiently applied or produce satisfactory solutions. Gupta and Sexton (1999) used standard BP, and extended delta-bar-delta variation in BP, and binary-coded GA for optimization of weight space for approximating chaotic time series. The authors showed that statistically superior solutions could be obtained from binary-coded GA. Ndiritu and Daniell, (2001) advised that in order to overcome the problems involved in the binary-coded GA, one needs to completely re-design the simple GA. The binary representation of continuous variables affects the efficiency and effectiveness of GA in finding the global optimal solution and creates problems in containing narrow feasible region (Deb and Agarwal 1995).

Chapter 3

Soft Computing Techniques

3.1 General

Since early 1960s, artificial intelligence (AI) has found its way into industrial applications mostly in the expert knowledge based decision making. In the past engineers have used a variety of tools for performing both causal modeling and inverse mapping. This set of tools includes statistics, optimization, rules of thumb and knowledge based systems, among others. The significance of soft computing came into limelight with the invention of fuzzy chips and the extensive applicability of Artificial Neural Networks (ANNs). Now at the beginning of the 21st century, soft computing plays a major role in modeling, inverse mapping, system identification, and control of systems – simple or complex (Zilouchian and Jamshidi, 2001). Generally, soft computing techniques include ANNs, fuzzy, and evolutionary computation. Soft computing paradigms and their hybrids are commonly used to enhance AI to incorporate human expert knowledge in computing process. Their application includes design of systems, control and management of water resources systems, and handling of complex systems with unknown parameters, such as prediction of industrial and natural processes, etc.

In this chapter, soft computing techniques and their hybrids used in the field of hydrology; namely ANN, and the Genetic Algorithm (GA), which will be used in the

current study, are discussed. The first section of this chapter presents the details of the ANNs, while the second section discusses the details of GA.

3.2 Artificial Neural Networks

Neural networks are mathematical models inspired by the functioning of biological neurons of human brain. The approximation to the biological neuron was first proposed by McCulloch-Pitts (1943). The *artificial neuron* is the basic building block/processing unit of a neural network called Artificial Neural Network (ANN). The origin of the idea of ANNs starts from perceptron developed by Rosenblatt in the late 1950s based on McCulloch-Pitts' conceptual computational element. Rosenblatt's perceptron model paved the way for both supervised and unsupervised algorithms seen today such as back propagation (BP) and self-organizing maps (SOM).

During the last decade, various ANN structures have been proposed by researchers in order to take dual advantage of ability of the human brain and computational speed of the computers. Given sufficient data, ANNs are well suited to the task of forecasting. ANNs excel at pattern recognition and forecasting from pattern clusters. In general, the major types of tasks in civil engineering to which ANNs are currently being applied are: classification and interpretation, diagnosis, inverse mapping, modeling, and control. In the field of hydrology, ANNs are applied at the level of weather system modeling, rainfall-runoff modeling, evaporation modeling, infiltration modeling, ground water modeling, and classification, etc. The present study concentrates mainly on the individual and integrated effects of different structures of ANNs, namely, Multi-Layer Feed-Forward Networks and the Self-Organizing Maps.

3.2.1 Multi-Layer Feed - Forward Neural Networks

The multi-layer feed-forward (FF) neural networks represent a generalization of the single layer perceptron. These can form arbitrarily complex decision regions and can separate various input patterns. The organization of neurons arranged in layers with no links between neurons in the same layer, no backward links, and no links skipping a layer are called multi-layer feed-forward neural networks. Thus, in the case of multi-layer feed-forward neural networks, the information flows, as indicated in Figure 3.1, only in the forward direction. Many studies proved that the multi-layer feed-forward neural networks perform as a class of universal approximators (Kolmogorov, 1957; Hecht-Nielsen, 1990; and Spreecher, 1993). The universality of single hidden layer neural networks in approximating any continuous function has been proved by many researchers (Cybenko, 1989; Hornik et al., 1989; and Funahasi, 1989).

In the field of hydrology, rainfall-runoff modeling is considered as a continuous functional mapping, where the ANNs interpret the problem as a cause-effect mapping. A key characteristic of this mode of operation is that problems in close proximity in the input space tend to map onto the solutions in close proximity in the output space (Flood and Kartam, 1997). Networks that incorporate neurons with continuous activation function generally provide this type of mapping. The structure of a single hidden layer feed-forward neural network (Figure 3.1) consists of interconnected neurons, each neuron having one or more incoming paths. Each incoming path has a signal on it (x) and the strength of the path is characterized by a weight (w). The neuron j sums the path weight times the input signal over all paths coming from previous layer neuron i; in addition, the neuron may be biased by an amount (b). Mathematically, the sum is expressed as:

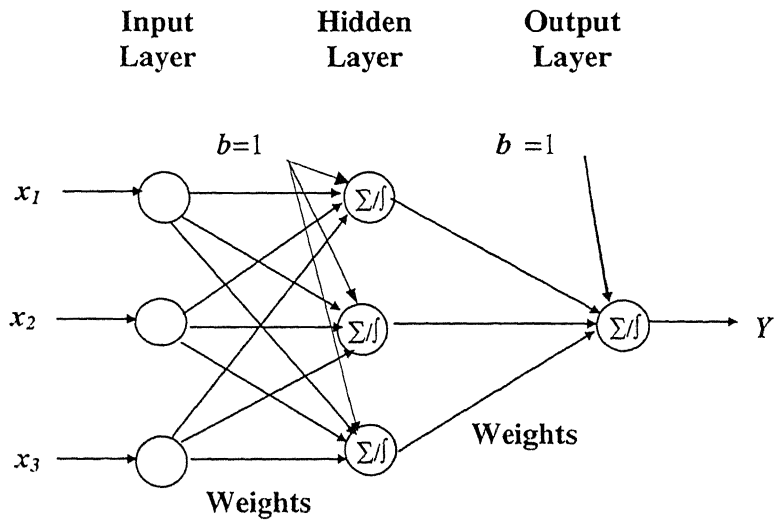


Figure 3.1: Structure of a Typical Feed-Forward Neural Network

$$Net_j = \sum_{i=1}^{n_l} w_{ij} x_i + b \tag{3.1}$$

where w_{ij} is the strength of connection from neuron i to neuron j, x_i is the output coming out from the neuron i, Net_j is the input received by neuron j, which is connected to neurons in a previous layer, n_l is the number of neurons in the previous layer than the one in which neuron j is located.

The output (y) of the each neuron is usually the logistic transformation of the sum using an activation function. In majority of the cases for modeling of rainfall-runoff process, sigmoid function has been used as activation function (Dawson and Wilby, 2001), which is continuous, bounded, non-decreasing, and differentiable. The sigmoid function maintains the value of activation within the bounds of 0 and 1. The use of such logistic

functions introduces non-linearity in the operation of the ANNs thereby underpinning and enhancing their ability to model non-linear processes. Mathematically, it can be described as:

$$y_j = f(Net_j) = \frac{1}{1+\exp(-Net_j)} \tag{3.2}$$

One of the most significant attributes of an ANN is its ability to learn by interacting with the information source. The input/output mapping of a network is established according to the weights and the activation functions of their neurons in hidden and output layers. The number of input variables corresponds to the number of input neurons in the ANN, and the number of output neurons is the same as the number of desired output variables. The number of neurons in the hidden layer depends upon the particular ANN application. The learning in an ANN is normally achieved through continuous updating of connection strengths through an adaptive procedure.

Training of Multi-Layer FF Networks

The learning process of a multiplayer feed-forward (FF) network can be one of the two kinds: supervised learning, or unsupervised learning. In supervised learning, the mechanism of updating the weights between the neurons as the patterns are presented to the ANN is controlled by external means based upon some theoretical concepts. In majority of ANN modeling applications of rainfall-runoff process, the training of multi-layer FF neural networks is performed using error back propagation (Dawson and Wilby, 2001). The extension of the gradient-based delta rule to multi-layer FF neural networks is the generalized delta rule popularly known as error back propagation (BP) algorithm (Reumelhart et al., 1986 a, b). In this, the error is back propagated to adjust

weights of the ANN. The learning process in multi-layer FF neural networks using back propagation algorithm is explained below.

The generalized delta rule (GDR) performs a gradient descent on the total sum of squares of errors of all the patterns over all neurons in the output layer. According to the BP algorithm the weight modification of a connection is proportional to the change in the total average error with respect to a unit change in the weight. In batch learning, the total error is cumulative error for all patterns and for all output neurons. To minimize the total error the weight changes should be in the negative gradient direction.

$$\Delta w_{ij} = -\eta_l \frac{\partial e_p}{\partial w_{ij}} \tag{3.3}$$

$$e_p = \frac{1}{2n} \sum_{k=1}^{n} (d_k - y_k)^2 \tag{3.4}$$

where e_p is the total average error for all the patterns and for all the output neurons, k is an index for neurons in the output layer, n is the number of neurons in the output layer, d_k is the desired output from neuron k in the output layer, y_k is the computed output from a neuron k in the output layer, and η_l is a learning coefficient. Equation 3.3 is called the weight updating equation, which can be rearranged as follows:

$$\Delta w_{ij} = -\eta_l \frac{\partial e_p}{\partial y_j} \frac{\partial y_j}{\partial w_{ij}} \tag{3.5}$$

The above equation can be written as

$$\Delta w_{ij} = \eta_l \, \delta_j \, y_i \tag{3.6}$$

where y_i is the output from the neuron i in the previous layer and input to the neuron j in the present layer, and δ_j is the error signal and is defined as follows:

(1) if j is an output layer neuron then:

$$\delta_j = y_j (1 - y_j)(d_j - y_j) \tag{3.7}$$

(2) if j is not an output layer neuron then:

$$\delta_j = y_j (1 - y_j) \sum_{k=1}^{n_2} \delta_k \, w_{jk} \tag{3.8}$$

Here, n_2 is the number of neurons in the layer next to the one in which neurons j is located. One of the drawbacks of the BP learning algorithm is the long duration of the training period. In order to improve the learning speed and avoid local minima, usually a momentum term is included in the weight updating equation. The updating equation then becomes:

$$\Delta w_{ij}(t+1) = \eta_l \, \delta_j \, y_i + \alpha \, \Delta w_{ij}(t) \tag{3.9}$$

Where α is the momentum coefficient and, t is the index for iteration. Therefore, the updated weights of the network then can be computed as follows:

$$w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij} \tag{3.10}$$

This way one iteration of error back-propagation and weight adjustments gets completed. This procedure is repeated until an acceptable level of convergence is achieved. This completes training of a multi-layer FF networks.

3.2.2 Self-Organizing Maps (SOM)

Self-Organizing Map (SOM) networks, proposed by Kohonen (1982), are also called Kohonen networks. The output from these networks is topologically ordered in the sense that the nearby neurons in the output layer correspond to a similar input. Additional features of Kohonen's map are the mapping from a continuous input space to a discrete output space. The self organization feature maps attempt to map a set of input vector x^k in N-dimensional input space on to an array of neurons, normally one or two dimensional, such that any topological relationship among $\boldsymbol{x}^k$ patterns are preserved and are represented by the network in terms of a spatial distribution of neuron activity. Selection of the number of output neurons determines the resolution of the output map. Kohonen's network ability to transform input relationships into spatial neighborhoods in the output neurons make interesting applications such as classification, feature mapping, feature extraction, etc. Thus, the SOM networks learn both the distribution and topology of the input vectors that they are trained on. Mappings from higher to lower dimensions are also possible with SOM and are, in general, useful for dimensionality reduction of input data. Self-organizing maps differ from other conventional competitive learning, where the SOM updates the weights of the winner and its neighbors.

Training of SOM Networks

The training of a SOM network is explicitly computed based on the design requirements in lieu of training. Self-organizing learning comes under unsupervised learning. The

learning is based on the clustering of input data. For the classification of input vectors, clustering is meant to be the grouping of similar objects and separating of the dissimilar ones. No priori knowledge is assumed to be available regarding the membership of an input in a particular class. Rather, gradually detected characteristics and a history of training are used to assist the network in defining classes and possible boundaries between them.

The SOM network used in this research classifies input vectors into one of the specified number of p categories according to the clusters detected in the training set. These can be treated as One-Dimensional Kohonen network. The training is performed in an unsupervised mode, using Kohonen's algorithm and the network undergoes the self-organization process. Figure 3.2 illustrates Kohonen's feature map, transforming two-dimensional data onto one-dimensional output space. A two dimensional input vector consists of (x_1, x_2), where x_1 and x_2 are normalized input patterns between [0,1] and the output is represented by neurons, represented as solid circles, and the coordinates of the neurons are the weights. The output neurons retain the neighborhood relations of the input space in the sense that (x_1, x_2), which forms a cluster, represent adjacent neurons in the output space arranged in a two-dimensional grid to form a vector quantizer. Input vectors are presented sequentially in time and after sufficient number of input vectors have been presented, weights specify clusters or vector centers. The clusters and vector centers sample the input space such that the point density function of the vector centers approximate the probability density functions of the input vectors. The training algorithm also organizes the weights so that topologically close neurons are sensitive to physically similar inputs. Output neurons are thus ordered in a natural fashion (Iyengar et al. 2002).

The objective of learning in the SOM networks in the present research is to classify or to decompose the input output space into different portions based on the kind and behavior of the neurons. The more related are two patterns in the input space, the closer one can expect the position in the array of the two neurons representing these patterns. This is the idea in developing a topographic map of the input vectors so that similar input vectors would trigger nearby neurons. Learning in a self-organizing feature maps occur for one vector at a time. The procedure to train a SOM network is as follows:

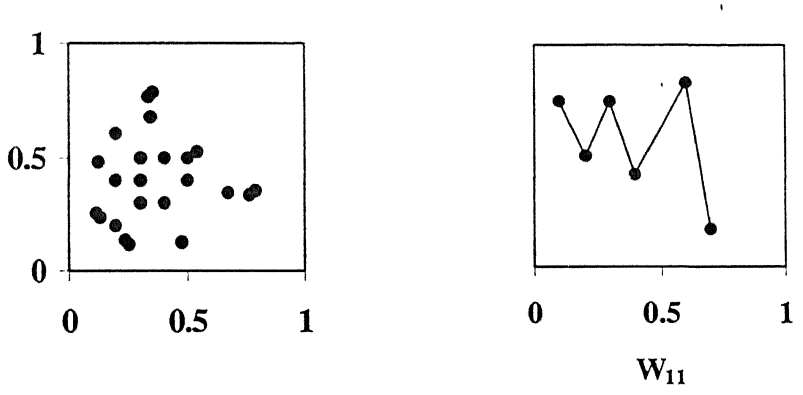


Figure 3.2: Two – Dimensional Data Mapped onto One-Dimensional space

1) The normal training starts by properly initializing the weight vector, normalization of weight vectors, and the input vectors.

2) The input pattern vector x ($x_1, x_2 \ldots . x_n$) is presented to the network, which defines a point in N-dimensional space. All the neurons in the output layer receive this input vector.

3) The winning neuron can be chosen based on competitive learning rule. Since, no information is available from outside as far as the desired classifier's responses, SOM uses the similarity of incoming patterns as the criterion for clustering. To define a cluster, one needs to establish a basis for assigning patterns to the domain of particular cluster. The most commonly used similarity rule is the Euclidean distance. The winning neuron i^* satisfies the condition that it will be the one with largest similarity measure between all weight vectors W_i and the input vector x. If the shortest Euclidean distance is selected as similarity measure within the cluster, then the winning neuron i* satisfies the following learning rule

$$\|W_{i^*} - x\| \leq \|W_i - x\| \qquad \text{for all } i \tag{3.11}$$

4) As the training progresses, the winning neighborhood region around winner neuron decreases. The radius of the winner neuron neighborhood region can be fairly large as the learning starts and slowly reduces to include only the winner and possibly its immediate neighbors.

5) The weight of winner neuron and its neighbors are updated by the Kohonen's rule, given by

$$\Delta W_i = \eta_l \; Nc\,(i,i^*)\,(x - W_i^{old}) \tag{3.12}$$

where η is the learning rate ($0 < \eta_l < 1$) and $Nc(i,i^*)$ is the neighborhood function such that $Nc(i,i^*) = 1$ if $i = i^*$ but falls off with distance $|r_i - r_i^*|$ between neurons i and i^* in the output array. The winner and closeby neurons are updated appreciably more than

those farther away. The typical choice for neighborhood function is Gaussian function given by

$$N_c(i,i^*) = e^{-\left(\left|r_i - r_i^*\right| / 2\sigma^2\right)} \tag{3.13}$$

The new weight is obtained by

$$W_i^{new} = W_i^{old} + \Delta W_i \tag{3.14}$$

Steps (3) to (5) are repeated after presenting each input vector until the training phase is completed for all the inputs.

In order to achieve a good convergence for the above procedure, the learning rate η, as well as the size of neighborhood N_c should decrease gradually with each iteration. The η_l is decreased for convergence, and σ is a parameter that is gradually decreased to contract the neighborhood. The Kohonen algorithm is divided into two phases: an ordering phase and a tuning phase. The learning rate, η_l, and the neighborhood size are also altered through these two phases.

The ordering phase lasts for given number of steps. In this phase, the neighborhood function is selected with an aim to allow neurons that respond to similar inputs to be brought together. In the beginning, the neighborhood distance and learning rate starts from maximum and decreases to the tuning neighborhood distance and to the tuning phase learning rate. The neighborhood radius may decrease linearly as time goes. The

learning rate is kept high to allow self-organization of input space onto the output space. The parameter η_l can also be adjusted linearly over time. Moreover, as the training progresses the network will typically order winning neuron and their neighbors in the input space with the same arrangement of neurons that they are ordered physically.

The tuning phase lasts for the rest of training. In this phase, the neurons are fine tuned to adjust to the distribution of the output map. However, the neighborhood distance stays at the tuning neighborhood distance and the learning rate continues to decrease very slowly from the tuning phase learning rate. However, in this phase the ordering learned in previous phase is stable. This way, the learning process in SOM networks is completed until all input patterns are presented to the network. The outcome of the training is the decomposition of the input output data space into a desired number of categories.

3.3 Genetic Algorithm

During last thirty years or so, there has been a growing interest in problem solving based on principles of evolution. One type of evolution system is called Genetic Algorithm (GA). The GA is a search technique based on the concept of natural selection inherent in natural genetics, which combines an artificial survival of the fittest with genetic operators abstracted from nature (Holland, 1975). The major difference between classical optimization search techniques and GA is that the GA works with a population of possible solutions rather than with a single solution. An individual solution in a population of solutions is equivalent to a natural chromosome. Like a natural chromosome completely specifies the genetic characteristics of a human being, an artificial chromosome in GA completely specifies the values of various decision

variables representing a decision or a solution. For most GAs, candidate solutions are represented by chromosomes; coded using either a binary number system or a real decimal system. The GA that employs binary strings as its chromosomes is called a *binary-coded* GA; whereas GA that employs real valued strings as its chromosomes is called a *real-coded* GA.

3.3.1 GA Operators

Regardless of the coding method used, the GA consists of three basic operators, selection, crossover or mating, and mutation. These are discussed in detail as follows:

(a) Selection: The GA starts with randomly generating an initial population of possible solutions (chromosomes). These chromosomes are evaluated based on their performances (fitness values) in terms of certain objective function. The selection of an individual to become a parent is primarily based on the fitness value of the chromosome. The better an individual's fitness value, the greater are its chance of being selected to be a parent. There are basically three types of selection algorithms: (1) proportionate selection, (2) linear rank selection, and (3) tournament selection. In proportionate selection, individuals are selected based on their fitness relative to all other individual in the population. In linear rank selection, the current population of individuals first sorted from best to worst by order of the fitness. Then each individual in the population is assigned a new fitness, based on applying a linear ranking function to the rank of the individual within the current population. Afterwards the selection of the parent will follow the proportionate selection method. The chromosomes compete for survival in a tournament selection, in which one parent is selected having the best fitness value among two or more randomly picked chromosomes. A second parent is selected by repeating the same process. This process of selection of individual

chromosomes based on their relative fitness is called natural selection. The mostly widely used type of tournament selection method is called binary tournament selection. The chromosomes compete for survival in a tournament selection, where the chromosomes with high fitness values enter the mating population and the remaining ones die off. The selected chromosomes form what is known as the mating population on which the crossover operator is applied.

(b) Crossover: The selection of parents for crossover in the mating pool is carried out with a probability of crossover P_c. A chromosome is randomly assigned a mating partner from within the mating population. In the binary-coded GA, the crossover is carried out by simply swapping the binary digits 0 or 1 at the random crossover location. The method of carrying out the crossover operator in binary-coded GA is shown in Figure 3.3, in which P1 and P2 represent the two parents, C1 and C2 represent the two children created after crossover, and the character '{' represents the random crossover location. However, in multi-variable engineering problems, each chromosome can represent more than one physical variable, and it may be possible to have a positional bias in the children towards certain variable(s) when the single point crossover is carried out as discussed above.

In a real-parametric space, the implementation of crossover operator to create new pair of offspring vectors is difficult when compared to the implementation of crossover operator on finite binary strings. Herrara et al. (1998) have provided a good overview of real-parameter crossover and mutation operators such as linear, naïve, BLX-α crossover, and random and uniform mutation etc. Deb and Agarwal (1995) developed Simulated Binary Crossover (SBX) operator, which simulates the principle of the single

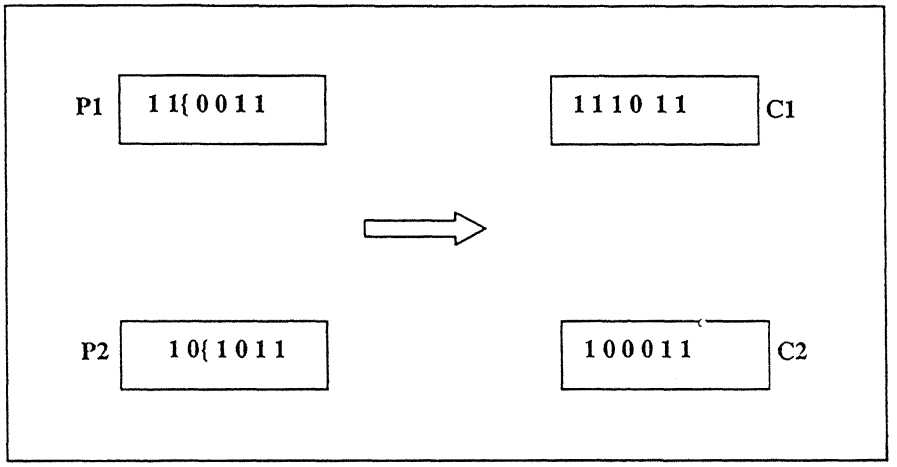


Figure 3.3: Single Point Crossover in Binary Coded Genetic Algorithm

point crossover to create offspring from the mating population of solutions. The procedure for computing offspring from two parents using SBX operator is explained here in brief. First, a random number (say u_i) between 0 and 1 is created. Then, from a specified probability distribution function, the ordinate $f(u_i)$ is found such that the cumulative probability for ordinate $f(u_i)$ is equal to u_i.

$$f(u_i) = \begin{cases} (2u_i)^{1/(\eta_c+1)} & \text{if } u_i \le 0.5, \\ \left(\dfrac{1}{2(1-u_i)}\right)^{1/(\eta_c+1)} & \text{otherwise,} \end{cases} \tag{3.15}$$

The specified probability distribution function, $f(u_i)$, involves a parameter (η_c), also called spread factor, which is a non-negative real number. The parameter η_c enables one to exercise certain degree of control in creating the children e.g. larger value of parameter (η_c) helps in creating 'near parent' solutions and smaller value of parameter

(η_c) helps in creating 'distant parent' solutions. After finding the ordinate $f(u_i)$, the offspring are calculated using the following equations:

$$C1 = 0.5\,[(1+f(u_i))\,P1 + (1-f(u_i))\,P2] \tag{3.16}$$

$$C2 = 0.5[(1-f(u_i))\,P1 + (1+f(u_i))\,P2] \tag{3.17}$$

The above procedure is used for variables where no lower and upper bounds are specified. Thus, the children solutions lie anywhere in the real space $[-\infty,\infty]$ with varying probability. For calculating children solutions where lower and upper bounds (P_L and P_U) of a parent are specified, the equation 3.16 can be modified as follows: (Deb, 1999)

$$f(u_i) = \begin{cases} (\alpha u_i)^{1/(\eta_c+1)} & \text{if } u_i \le \dfrac{1}{\alpha}, \\ \left(\dfrac{1}{(2-\alpha u_i)}\right)^{1/(\eta_c+1)} & \text{otherwise,} \end{cases} \tag{3.18}$$

where $\alpha = 2-\beta^{-(\eta_c+1)}$ and β is calculated as follows:

$$\beta = 1 + \frac{2}{C2-C1}\,Min[(P1-P_L),(P_U-P2)] \tag{3.19}$$

It is assumed here, that P1 < P2. A simple modification to the above equation can be made for P1 > P2. The above procedure allows a zero probability of creating any children solutions outside the prescribed range [X_L, X_U].

Like the single point crossover in binary-coded GA, the SBX operator is also applied with a probability of crossover of P_c. Thus, the SBX operator simulates the principle of single point crossover and also allows a certain degree of control in creating children in the desired vicinity of the parents. Further, in order to avoid the positional bias in the SBX operator, the SBX operator applied uniformly over all the variables with 50% probability, which is similar to the uniform crossover for multiple variables (Deb and Agarwal, 1995). Another advantage of using the SBX operator in *real-coded* GA is that in subsequent generations, the population is not guaranteed to be confined to a given range, because SBX operator has nonzero (albeit small) probability of creating children points far away from the parent points. This feature in *real-coded* GA may be more suitable in the problems in which the range of the decision variables in not known, which is the case in training of an ANN as the range of the weight matrix is unknown. This feature of the SBX operator is redundant for problems where the lower and upper limits of the decision variables are absolutely known. This feature of the SBX operator may help in the search for the global optima in a better way as compared to the BP algorithm. However, it needs to be investigated especially in reference to the hydrologic systems modeling.

(c) Mutation: The generation of children formed after the crossover operator has been carried out is expected to have high fitness values as they have been formed by mating of the two parents with high fitness values. If only selection and crossover operators are used in a GA, then it is possible for GA to converge to a local optimum. This is because of the fact that the GA is a very aggressive search technique and rapidly discards chromosomes with poor fitness values. In order to maintain diversity in a population from one generation to the next, a mutation operator is normally applied. In binary-

coded GA, mutation is achieved through a local perturbation (i.e. replacing 0 with 1 or vice-versa) in the binary strings, with a probability of P_m. The procedure of creating a mutated child from a parent using the parameter based mutation operator (Deb, 2001) is similar to the SBX operator in implementation, where the parent is perturbed by a specified amount. First, a random number (say r_i) between 0 and 1 is created. Then, from a specified probability distribution function, the ordinate $f(r_i)$ is found such that the cumulative probability for ordinate $f(r_i)$ is equal to r_i.

$$f(r_i) = \begin{cases} (2r_i)^{1/(\eta_m+1)} - 1 & \text{if } r_i \le 0.5, \\ 1-[2(1-r)]^{1/(\eta_m+1)} & \text{otherwise,} \end{cases} \tag{3.20}$$

The probability distribution used to compute the perturbation, involves a parameter (η_m), which controls the shape of the distribution. After computing $f(r_i)$, the offspring is calculated using the following equation:

$$C = P + (P_U - P_L) f(r_i) \tag{3.21}$$

Where C is the mutated child, P is a parent; P_U and P_L are the upper and lower bounds of the parent, respectively. The above procedure is used for variables where lower and upper boundaries are not specified. If the variable boundaries are specified, the above equations may be changed as follows.

$$f(r_i) = \begin{cases} [2r_i + (1-2r_i)(1-\delta)^{\eta_m+1}]^{1/(\eta_m+1)} - 1 & \text{if } r_i \le 0.5, \\ 1-[2(1-r) + 2(r_i - 0.5)(1-\delta)^{\eta_m+1}]^{1/(\eta_m+1)} & \text{otherwise,} \end{cases} \tag{3.22}$$

where $\delta = \min [(P - P_L), (P_U - P)] / (P_U - P_L)$. This ensures that no solution would be created outside the range $[P_L, P_U]$.

This process of selection, crossover, and mutation is repeated for many generations with the objective of reaching the global optimal solution after a sufficient number of generations. However, using a floating-point representation is conceptually closest to the problem space and also allows for an easy and efficient implementation of closed and dynamic operators. In particular, the parameter optimization problems with variable over continuous domain, we may experiment with *real-coded* genes together with special *genetic* operators developed for them (Michalewicz, 1996).

3.3.2. Advantages of *Real-Coded* GA

The binary representation traditionally used in GAs has certain drawbacks when applied to multi-dimensional, high precision numerical problems. Following are some of the limitations of binary-coded GA that can be overcome using *real-coded* GA (Deb and Agarwal, 1995; Michalewicz, 1996; and Deb, 2001).

1) The binary-coded GA actually discretizes the search space, which may be continuous in nature. In fact, in most of the engineering applications, the decision variables involved are real-valued and involve a continuous search space. Sometimes, this discrete representation of a continuous search space may lead to some problems *e.g.* search getting stuck in local optima. The *real-coded* GA allows the search for an optimal solution in a continuous real-valued search space.

2) The length of the binary-coded string increases for increased precision. Hence, for the problems requiring high precision, the lengths of the binary-coded strings need to be large resulting in an increase in the size of the required population (Goldeberg *et al* 1992), which in turn increases the computational complexity and reduces the speed of the algorithm. On the other hand, high precision can be easily achieved in the *real-coded* GA without having to increase the size of the population.

3) The binary-coded GA involves a single point crossover site along the length of a string. In multi-variable optimization problems, a binary string can represent more than one variable by representing a single variable as a subset of a larger string. The single point crossover may cause a large positional bias for certain combination of variables while propagating from parents to children. This problem is overcome in the *real-coded* GA by having a separate probability distribution that allows at least 50% of the variables in the chromosome of real variables and eliminates the possibility of any positional bias while propagating from parents to children.

4) Another limitation of a binary-coded GA is in a phenomenon called '*Hamming Cliff*', which poses hindrances while searching for a solution in the continuous search space. A *Hamming Cliff* is a region close to optimum solution where a lot of points (or solutions) lie in a population. When the Euclidean distance between two feasible solutions in a Hamming Cliff is small, then there is less likelihood of getting a better solution and vise-versa. Because of the high precision capability of the *real-coded* GA, it is possible to have two feasible solutions within the same Euclidean distance in a Hamming Cliff region as

compared to that in a binary-coded GA. This flexibility of the *real-coded* GA allows searching for a better solution as compared to the binary-coded GA.

5) When one is encountered with a tiny *feasible* region while searching for an optimal solution, the binary-coded GA (which depends on a single point crossover) may not be able to create *feasible* children from two feasible parent solutions. Once feasible parent solutions are found, a controlled crossover operator is desired in order to create children solutions that are also *feasible.* The floating-point representation of variables and the crossover parameter η_c in *real-coded* GA allow a little control over the crossover operation and hence the *feasibility* of the children solutions.

6) Because the binary-coded GA involves coding of the physical variables, each function evaluation also involves coding and decoding of the variables for computing the fitness values of the objective function. In multiple-variable complex optimization problems involving a large number of variables, this can be computationally not very efficient. However, because the *real-coded* GAs use decimal system, no time is lost in coding-decoding.

7) One has to specify lower and upper bounds of each physical variable in solving an optimization problem using GAs. In binary-coded GAs, such bounds are rigid but the *real-coded* GA with SBX crossover operator allows flexible boundaries because of the probability distributions used in crossover and mutation operators. The flexible bounds allow the search in a wider space increasing the chances of obtaining the global optimum solution.

3.3.3 Elitism in *Real-Coded* GA

The above section, presented the details of *real-coded* GA, which do not use any elitism in the algorithm. As the name suggests, the 'elitism' in the *real-coded* GA favors the *elites* of a population by giving them an opportunity to be directly carried over to the next generation. In the optimization problems, elitism is introduced into a GA in various ways: either at the level of creation of offspring solutions after the crossover and mutation operators, or globally in a generational sense.

No matter how elitism is introduced, it makes sure that the fitness value of the population's best solution does not deteriorate. In this way, a good solution found early on in the run will never be lost unless a better solution is discovered. The absence of elitism does not guarantee this aspect. In fact, Rudolph (1996) has proved that GAs converges to the global optimal solution of some functions in the presence of elitism. Moreover, the presence of *elites* enhances the probability of creating better offspring. The elitism considered in *real-coded* GA to enhance its efficiency and effectiveness is nothing but keeping a single best solution in the previous generation to replace with the worst solution of the current generation if the best solution of the current generation is worse than the best solution of previous generation. The *real-coded* GA source code developed by Kanpur Genetic Algorithm Lab (KanGAL, http://www.iitk.ac.in/kangal/) was modified to include elitism, and redesigned to suit optimization of parameters of the conceptual model and for training of an ANN. The tournament selection procedure, SBX crossover operator, and the parameter based mutation operator discussed above were implemented in the present study. The flow chart of the *real-coded* GA program that incorporates elitism is shown in Figure 3.4.

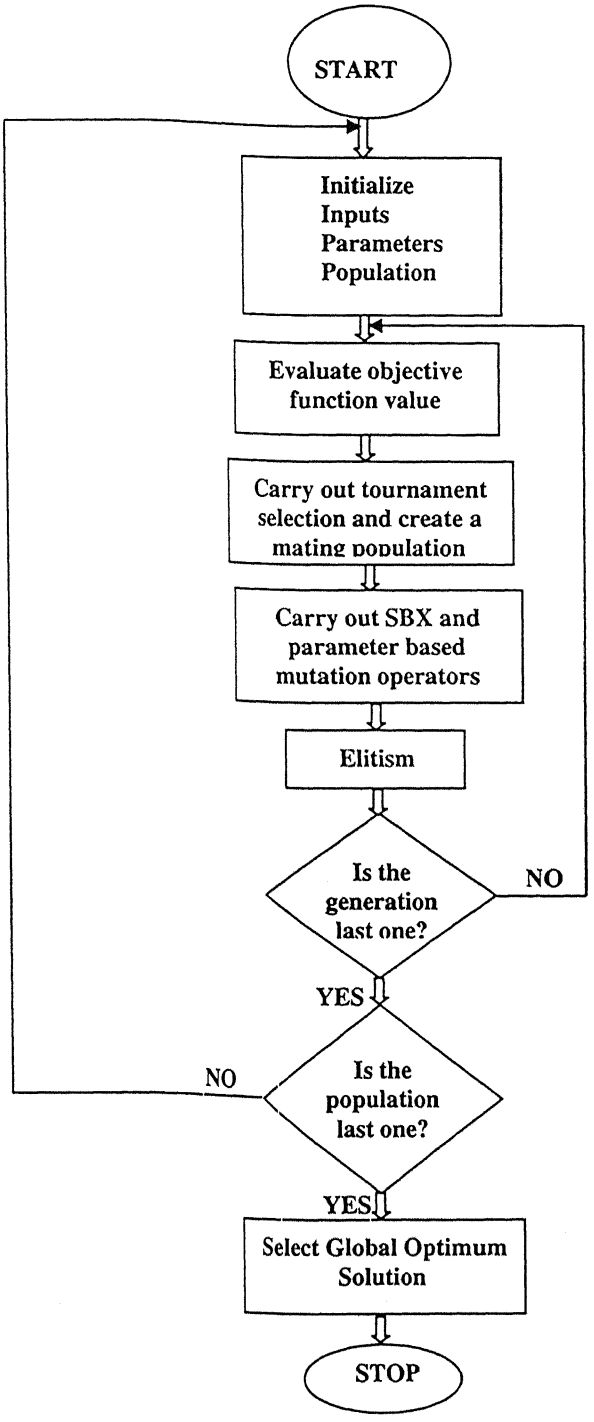


Figure 3.4: Flow Chart of Elitist *Real-Coded* GA

Chapter 4

Development of CRR Models

4.1 General

This study evaluates the techniques available for the modeling of the complex, dynamic, non-linear, and fragmented rainfall-runoff process. Specifically, two types of techniques are being investigated: conceptual techniques and soft computing techniques. Further, a separate category of rainfall-runoff models is explored that combines the conceptual and soft computing techniques to take advantages of both the techniques. This chapter discusses the conceptual rainfall-runoff (CRR) models developed in this study; whereas, the next chapter describes the modeling of rainfall-runoff process, using both soft computing techniques, and a combination of conceptual and soft computing techniques.

A CRR model is the one in which, instead of using the equations of mass, energy, and momentum to describe the process of water movement, a simplified, but plausible conceptual representation of underlying physics is adopted. These representations frequently involve several inter-linked storages and simplified budgeting procedures, which ensure that at all times a complete mass balance is maintained among all inputs, outputs, and inner storage changes. The rainfall-runoff process is a complex and non-linear process affected by many physical factors. The factors affecting the runoff response of a watershed include (a) storm characteristics *i.e.* intensity and duration of

rainfall events, (b) watershed characteristics *i.e.* storage characteristics of the watershed, size, shape, slope, of the watershed, and the percent of the watershed contributing runoff at the outlet at various time steps during a rainfall event *etc.*, (c) geo-morphological characteristics of a watershed *i.e.* topography, land use patterns, vegetation, soil types, *etc.* that affect the infiltration losses from the watershed, and (d) climatic characteristics such as temperature, humidity, and wind characteristics.

There are many CRR models with varying degree of sophistication and complexity reported in literature that attempt to model the complex, dynamic, non-linear, and fragmented rainfall-runoff process. Any CRR model involves two major steps, while attempting to model the process of transformation of rainfall into runoff (i) calculation of infiltration and other losses and estimation of the effective rainfall, and (ii) subsequent transformation of the effective rainfall into runoff through an operator which simulates the behaviour of the watershed being considered. The second step is actually responsible for modelling the non-linear, dynamic, and complex nature of the rainfall-runoff process. During this process, the input (i.e. effective rainfall) to the system (i.e. watershed) goes through two operators: (i) 'translation' in time due to variable source areas of the watershed contributing runoff at the outlet at different times, and (ii) 'attenuation' due to the storage characteristics of the watershed. All CRR models of the rainfall-runoff process essentially attempt to account for these two operators with varying degree of sophistication and complexity.

A simplified representation of the complex hydrologic process can be depicted in a schematic shown in Figure 4.1. The hydrologic system shown in Figure 4.1 consists of two major components: a surface flow component and a sub-surface flow component.

In Figure 4.1, P represents total rainfall, ER represents effective rainfall, F represents actual incremental infiltration, ET represents the expected evapotranspiration, QS represents surface flow component, QG represents sub-surface flow component, and Q represents the total response from the hydrologic system in the form of total discharge at the outlet of the watershed.

In this study, two types of CRR models have been developed, which conceptualize the hydrologic system to be made up of two storage elements: a surface storage component and a sub-surface storage component (see Figure 4.1), employ a continuous time domain infiltration equation to estimate infiltration at each time step, use the concept of flow recession to model the falling limb of a flow hydrograph, and use the law of conservation of mass to continuously update the sub-surface storage component. The following sections discuss the structure of the two CRR models.

4.2 CRR Model-I

The schematic of the hydrologic system employed in CRR Model-I is shown in Figure 4.2. On the rising limb of the hydrograph the total flow at the outlet of the watershed is composed of surface flow component and the base flow component. The total spatially aggregated rainfall, represented by P, is considered as input to the hydrologic system, part of the total rainfall input infiltrates into soil, and appears at outlet as a base flow after passing through subsurface storage. The remaining part of the total rainfall, as an effective rainfall (ER), runs through the surface store and appears at outlet as surface flow. The Green-Ampt infiltration equations are employed to compute the infiltration, then the effective rainfall is evaluated by subtracting infiltration from the total rainfall. The CRR Model-I consists of the following components: (a) base flow component,

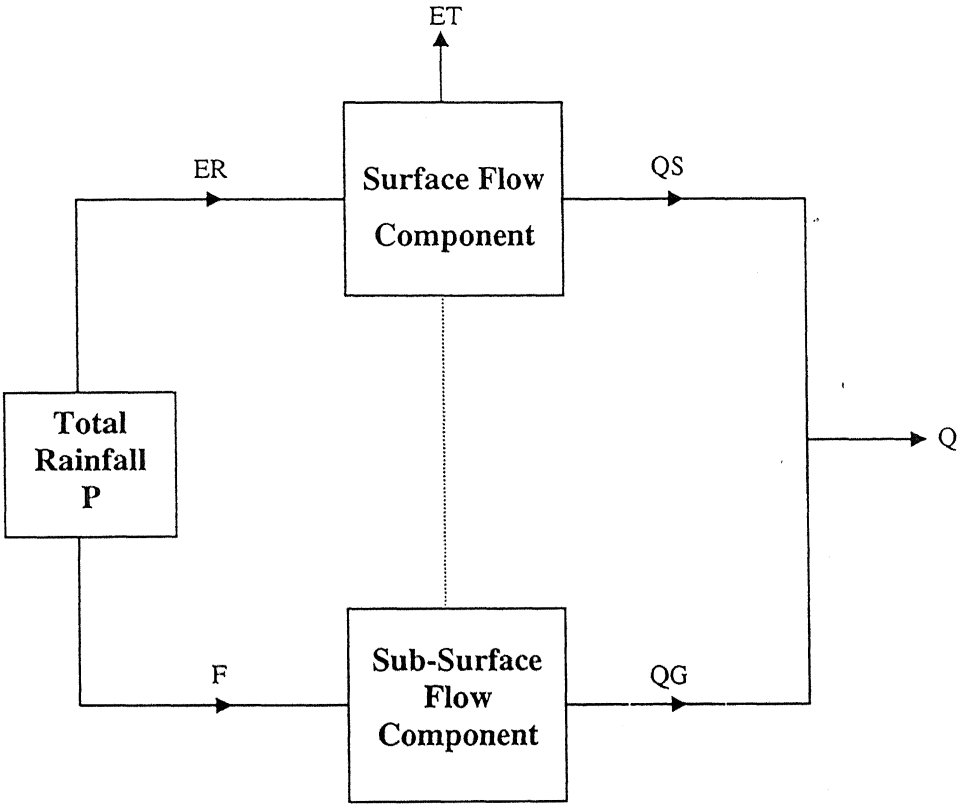


Figure 4.1: Schematic of the Simplified Hydrologic System

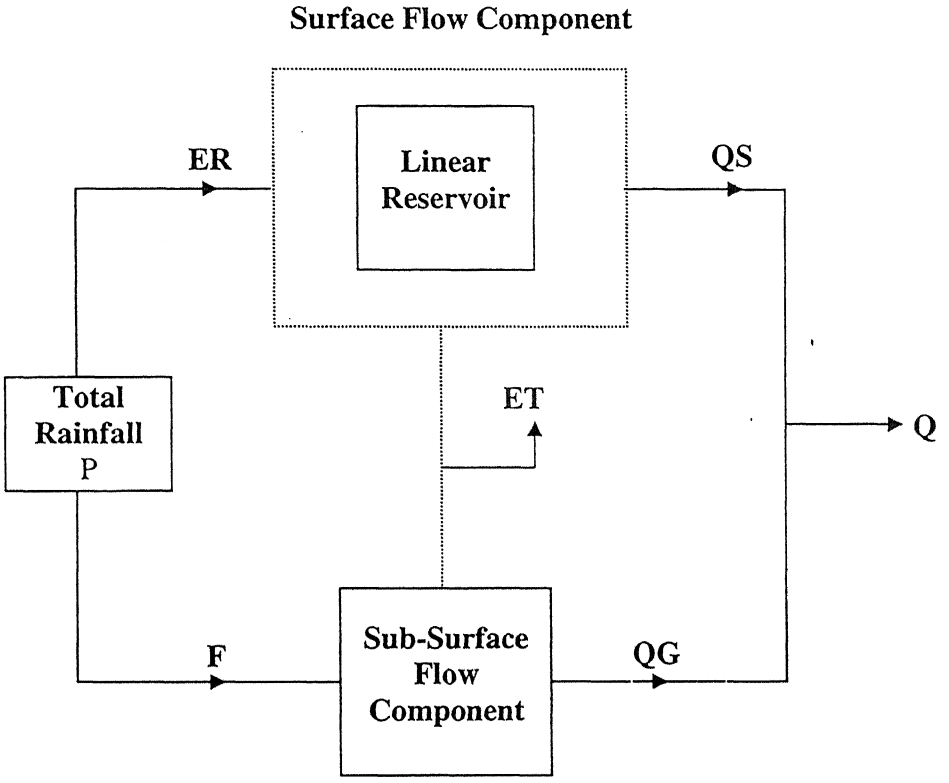


Figure 4.2: Schematic of the CRR Model - I

(b) infiltration component, (c) soil moisture accounting (SMA) component, (d) evapotranspiration component, and (d) surface flow component. These are discussed in details in the following sections. In the surface flow component the CRR Model-I assumes the watershed to be a linear reservoir with constant coefficients.

4.2.1 Base Flow Component

The sub-surface flow component at time t can be computed using the concept of base flow recession through the use of the following equations:

$$QG_t = KG_t * QG_{t-1} \tag{4.1}$$

$$KG_t = \frac{QG_{t-1}}{QG_{t-2}} \tag{4.2}$$

Where QG_{t-1} is the base flow at time t-1 in m^3/sec, QG_{t-2} is the base flow at time t-2 in m^3/sec, and KG_t is the recession coefficient for base flow at time t. The value of KG_t at the beginning of a flow hydrograph can be computed using the observed flow data of the previous flow hydrograph. The base flow at the beginning of a flow hydrograph is estimated by using the recession of the previous flow hydrograph as depicted in Figure 4.2. The base flow component can be updated as soon as the observed flow hydrograph starts to rise.

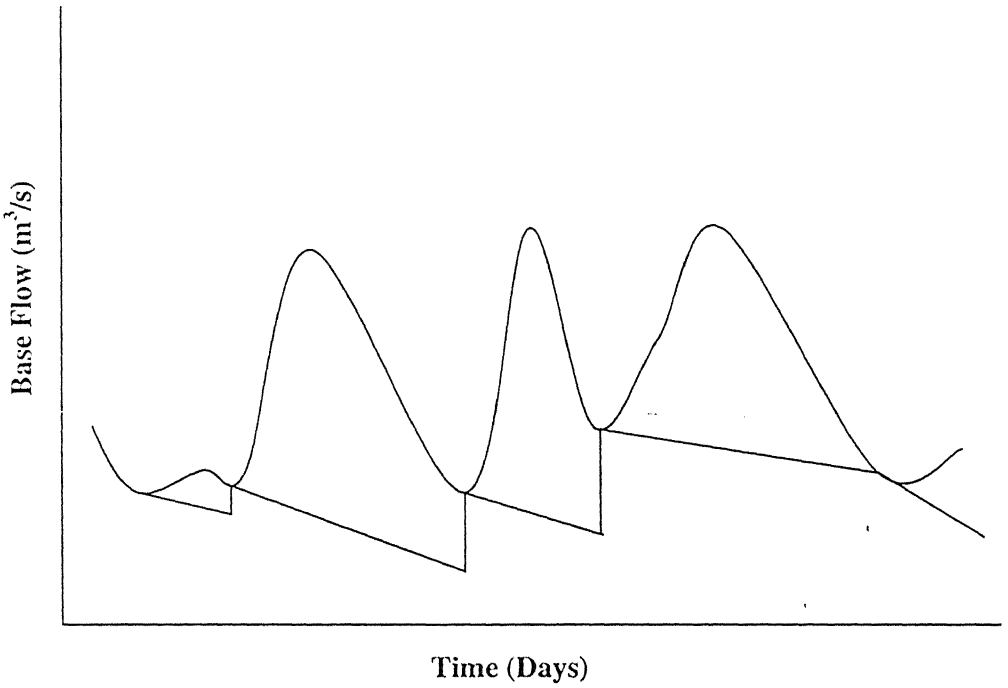


Figure 4.3: Base Flow Model Component in CRR Models

4.2.2 Infiltration Component

Any CRR model on a continuous basis needs computation of effective rainfall at each time step, which in turn would require the estimation of infiltration continuously. Once actual incremental infiltration at any time step '*t*' has been determined, effective rainfall can be computed using the following equation:

$$ER_t = P_t - \Delta FA_t \qquad (4.3)$$

Where, ER_t is the effective rainfall at time t in mm, P_t is the total rainfall input at time t in mm, ΔFA_t is the amount of actual incremental infiltration at time t in mm calculated using infiltration model.

A continuous time domain infiltration equation is needed for estimating infiltration at each time t. The Horton's and Green-Ampt's infiltration equations are the most commonly used conceptual methods, which provide estimates of the infiltration capacities as a function of time. Horton's formula is a conceptual model and the Green-Ampt formula is the exact analytical solution to an approximate physically based model (Hsu *et al.* 2002). It has been pointed recently that the Green-Ampt equations fit better than the Horton's formula in terms of the infiltration rate curves (Hsu *et al.*, 2002). In the context of continuous daily rainfall-runoff modeling, the updating of soil moisture at every time step is important while modeling the infiltration. The Green-Ampt method of computing infiltration allows use of updated soil moisture storage at every time step to determine potential infiltration, and was therefore adopted in the present study as the infiltration model.

The Green-Ampt infiltration equations were employed in this study to compute potential incremental infiltration continuously. The Green-Ampt infiltration equations can be described by the following set of equations.

$$f = K\left(\frac{\psi\Delta\theta}{F_t}+1\right) \tag{4.4}$$

$$F_t = K\Delta t + F_{t-1} + \psi\Delta\theta \ln\left[\frac{F_t+\psi\Delta\theta}{F_{t-1}+\psi\Delta\theta}\right] \tag{4.5}$$

$$\Delta F_t = K\Delta t + \psi\Delta\theta\ln\left[\frac{F_t+\psi\Delta\theta}{F_{t-1}+\psi\Delta\theta}\right] \tag{4.6}$$

Where

$$\Delta\theta = (1 - S_e)\theta_e \tag{4.7}$$

$$S_e = \frac{\theta - \theta_r}{\eta - \theta_r} \tag{4.8}$$

where f is the potential infiltration rate in mm/hr; K is the saturated hydraulic conductivity of the soil matrix in mm/hr; ψ is the soil suction head in mm; F_t is the cumulative infiltration into the soil storage up to a time step t in mm; S_e is the effective saturation(varies between 0 and 1); θ is the moisture content expressed as dimensionless fraction; θ_e is the effective porosity (η-θ_r); θ_r is the soil moisture retention; and η is the porosity of the soil. If θ_r is negligible and taking $\theta = (S_t/S)\eta$, the equation 4.8 becomes

$$S_e = \frac{S_t}{S} \tag{4.9}$$

where S_t is the updated subsurface store at the end of time t in mm and S is the maximum depth of the soil horizon in mm.

4.2.3 Soil Moisture Accounting (SMA) Component

An important step in using Green-Ampt infiltration equations to compute actual incremental infiltration at each time step, to be used in a rainfall-runoff model on a continuous basis, is to determine the initial soil moisture storage at the beginning of a storm corresponding to the current hydrologic and climatic conditions such as antecedent precipitation and base flow etc. Once determined, this can be updated continuously throughout the storm using a simple mass balance relationship. The

methodology for derivation of initial soil moisture storage at the beginning of each storm is given below. The determination of soil moisture storage at the beginning of a storm can be accomplished through the use of a simple regression model. The structure of this simple regression model can be described by the following equation:

$$SA_{t-1} = \beta_0 + \beta_1 QG_{t-1} \tag{4.10}$$

Where β s' are the regression coefficients to be determined, SA_{t-1} is the initial available soil moisture storage at time t-1 in mm at the beginning of a storm corresponding to the prevailing hydrologic and climatic conditions, and t is an index representing time. Once the available soil moisture storage is known, the soil moisture storage can be determined by subtracting SA_{t-1} from the maximum depth of soil moisture storage S. Once the initial soil moisture storage corresponding to the current hydrologic and climatic conditions has been determined, it can be updated continuously throughout and after the storm, using the following mass balance equation:

$$S_t = S_{t-1} + \Delta FA_t - ET_t - QG_t \tag{4.11}$$

Where S_t is the soil moisture storage at the end of time t in mm, ΔFA_t is the actual incremental infiltration during time interval Δt in mm, ET_t is the actual expected evapotranspiration at time t in mm, and QG_t is the base flow at the end of time t in mm. In current study, the actual daily evapo-transpiration was computed using the procedure proposed by Haan (1972) for the state of Kentucky discussed next.

4.2.4 Evapotranspiration Model Component

Hann (1972) has computed mean monthly potential evapotranspiration for the state of Kentucky (Table 4.1) by considering monthly evapotranspiration as a function of average monthly temperatures. Daily potential evapotranspiration is calculated by dividing the monthly potential evapotranspiration by the total number of days in a month. Then the actual evapotranspiration on a day t is computed by comparing known potential daily evapotranspiration and total daily rainfall as follows (Hann 1972).

$$ET = \frac{1}{2} ET_p \qquad \text{if } P \geq 2.5 \ mm/day \qquad (4.12)$$

$$ET = ET_p \qquad \text{if } P < 2.5 \ mm/day \qquad (4.13)$$

where, ET is the actual daily evapotranspiration at time t (mm/day), ET_p is the potential daily evapotranspiration at time t (mm/day), and P is the total rainfall on a day (mm/day).

4.2.5 Surface Flow Component

The total observed flow at the outlet of a watershed is the sum of surface flow and sub-surface flow components. This can mathematically be represented as follows:

$$QO_t = QS_t + QG_t \qquad (4.14)$$

Where QO_t is the observed flow from the system at time t in m^3/sec, QS_t is the surface flow component from the system at time t in m^3/sec, and QG_t is the base flow from the

Table 4.1: Monthly and Daily Potential Evapotranspiration due to Haan (1972)

Month	PET in mm/month	PET in mm/day
January	0.79	0.03
February	2.17	0.08
March	19.69	0.64
April	48.00	1.60
May	95.28	3.07
June	133.35	4.45
July	149.61	4.83
August	134.65	4.34
September	97.54	3.25
October	51.97	1.68
November	18.29	0.61
December	3.15	0.10

system at time t in m^3/sec. The surface flow (QS_t) is derived from the surface storage component using linear reservoir model. The inflow (ER_t) to the system can be routed through the linear-reservoir to compute QS_t using the following routing equation (Singh, 1988):

$$QS_t = C_1\left(\frac{ER^*_{t-1} + ER_t}{2}\right) + C_2\, QS^*_{t-1} \tag{4.15}$$

$$K_1 = \frac{Q_{t-2}}{\Delta Q_{t-2}} \tag{4.16}$$

$$\Delta Q_{t-2} = \frac{Q_{t-1} - Q_{t-3}}{2} \tag{4.17}$$

$$C_1 = \frac{2\Delta t}{2K_1 + \Delta t} \tag{4.18}$$

$$C_2 = \frac{2K_1 - \Delta t}{2K_1 + \Delta t} \tag{4.19}$$

where C_1 and C_2 are the linear reservoir routing coefficients, ER^*_{t-1} is the updated amount of inflow coming into the surface store at time $(t\text{-}1)$, ER_t is the actual amount of inflow coming into the surface store at time t, QS^*_{t-1} is the updated amount of surface flow component at time t-1, K_1 is an attenuation constant determined using Clark's method form historical flow data, and Δt is the time interval of the model. The QS^*_{t-1}

can be computed using the observed stream flow and base flow in the past, using the following equation:

$$QS^{*}_{t-1} = QO_{t-1} - QG_{t-1} \tag{4.20}$$

where, QO_{t-1} is the observed streamflow at time step (t-1) in m^3/sec and QG_{t-1} is the computed base flow at a time step (t-1) in m^3/sec. Further, the updated value of ER^{*}_{t-1} to be used in the reservoir routing equation can be estimated by using the reservoir routing equation in inverse direction written at a previous time step as follows:

$$ER^{*}_{t-1} = 2\frac{(QS^{*}_{t-1} - C_2\, QS^{*}_{t-2})}{C_1} - ER^{*}_{t-2} \tag{4.21}$$

The falling limb of the flow hydrograph can be modelled by using adaptive decay model, which can be mathematically expressed as follows:

$$Q_t = Kf_t\ QO_{t-1} \tag{4.22}$$

where Q_t is the modelled streamflow at time t on falling limb in m^3/sec, QO_{t-1} is the observed streamflow at time t-1 on falling limb in m^3/sec, and Kf_t is the decay coefficient at time step t. The value of decay coefficient can be as follows

$$Kf_t = \frac{QO_{t-1}}{QO_{t-2}} \tag{4.23}$$

where, QO_{t-2} is the observed stream flow on the day t-2 on the falling limb in m^3/sec. The overall model structure of the CRR Model–I can be summarized in the following steps.

1. Compute the daily average rainfall using arithmetic mean method, and actual daily evapotranspiration following Haan's procedure.
2. Compute the base flow using equation 4.1
3. Compute the constant linear reservoir routing coefficients C_1 and C_2 (equations 4.16 through 4.19) using observed streamflow data from previous time steps.
4. Compute the initial available soil moisture using equation 4.10 and then compute initial soil moisture storage.
5. Compute the potential incremental infiltration amount at a time step t (ΔF_t) using the equation 4.6, actual incremental infiltration at time t (ΔFA_t) by comparing ΔF_t with P_t, and then the effective rainfall using equation 4.3.
6. If the streamflow is on the rising limb then compute the updated inflow component (ER^*_{t-1}) using the equations 4.21, and compute the updated surface flow component (QS^*_{t-1}) using equation 4.20. Then compute surface flow component (QS_t) using equation 4.15 and add it to base flow to predict the streamflow.
7. If the streamflow is on the falling limb then use decay model (equation 4.22 and 4.23)
8. Update all parameters and stores using appropriate equations described above.

4.3 CRR Model–II

The CRR Model–I developed in this study used a linear reservoir to model the rainfall-runoff process. Also, the parameters of the routing model were constant. Because the rainfall-runoff process is highly non-linear, complex, and dynamic in nature, the CRR Model–I may not be able to capture such a relationship. Therefore, in an attempt to improve upon the structure of the CRR Model-I, a second CRR model (called CRR Model-II) was developed. The CRR Model-II was same as CRR Model-I as far as infiltration, evapotranspiration, base flow, soil moisture accounting model components are concerned. The CRR Model-II differed from CRR Model-I in the surface flow component. The surface flow component is represented by two conceptual hydrological elements. The first element is a linear channel in the form of a time-area diagram for linear translation of the rainfall input. The output from the linear channel element is then routed through the second element in the form of a non-linear reservoir to account for the non-linear & dynamical attenuation effects of the watershed. The CRR Model–II had the capability to continuously update the routing coefficients adaptively using the recently observed discharges.

Thus, the salient features of the CRR Model-II are that it (a) conceptualises the watershed as a non-linear reservoir, (b) employs a continuous time domain infiltration equation (Green-Ampt) to compute infiltration and hence effective rainfalls, (c) convolutes the effective rainfalls through a time area function of a watershed to compute convoluted effective rainfalls, (d) routes the convoluted effective rainfalls through the watershed represented by a non-linear reservoir to model the surface flow component, (e) models the base flow component using the concept of base flow recession, and (f) has the capability to update all the model parameters adaptively and update the sub-

surface component continuously using the principle of mass-balance. The structure of the CRR Model-II having the capabilities (a) through (f) described above, is shown in Figure 4.4, which is capable of capturing the non-linear and dynamic nature inherent in a complex rainfall-runoff process. The total streamflow derived from the rising limb of the hydrograph is the sum of surface and base flow component. Stream flow on the falling limb of the hydrograph was derived using an adaptive decay model as described in the CRR Model-I.

4.3.1 Surface Flow Component

The flow contribution (QS_t) was modelled in CRR Model-II to account for translation and attenuation effects of the watershed using combination of linear channel and non-linear reservoir components.

4.3.1.1 Linear Channel Component

The time area method is important from the point of view of time distribution of rainfall on runoff. The central idea in the time area method is of a time contour or an *isochrone*. An isochrone is a contour joining those points in the watershed that are separated from the outlet by the same travel time. Transforming effective rainfalls through a time area diagram to get convoluted effective rainfalls may be thought of as a linear channel. The steps involved in the construction of a time area diagram (after Singh, 1988) are described below.

1. Compute the time of concentration (T_c) for the watershed under the investigation. This can be determined by analyzing the observed data, such as, precipitation in the watershed and observed peak streamflows at the outlet.

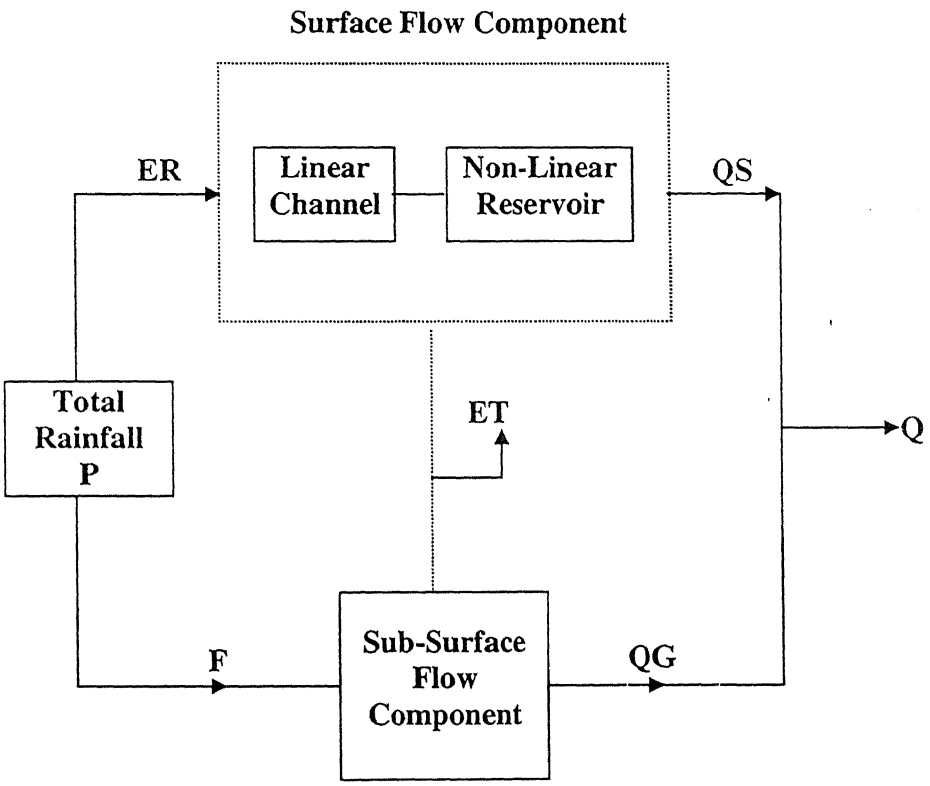


Figure 4.4: Schematic of the CRR Model-II

2. Select an appropriate time interval and divide T_c by it. The time interval was selected to be one day, as it was also the time horizon for the conceptual rainfall-runoff models.
3. Construct isochrones on the watershed map at a spacing equal to the time interval chosen in step 2. This can be accomplished by plotting a profile of the main channel, and using the contour lines on the topographic map.
4. Compute the area between the isochrones. These areas at different time intervals give the ordinates of the time area concentration (TAC) diagram.
5. Compute the cumulative areas at various time intervals to obtain the ordinates of the time area (TA) diagram.

A time area diagram was constructed for the Kentucky River watershed. The time of concentration for the Kentucky River watershed was determined to be two days. The ordinates of the time area (TA) diagram, and time area concentration (TAC) diagram are given in the Table 4.2. Using the ordinates of the 'TAC' diagram, the effective rainfall can be convoluted to compute runoff, according to the following equation;

$$Q_j = C_u \sum_{i=1}^{j} ER_i \, a_{j-i+1} \tag{4.23}$$

Where Q_j is the translated output from the linear channel in m^3/sec, a is the ordinate of the TAC diagram in km^2, and ER_i is the effective rainfall in mm/day. C_u is the constant accounting for units (=0.011574). The runoff thus computed was converted into a convoluted effective rainfall depth that was then routed through the non-linear tank model to compute the surface flow component, which was discussed next.

Table 4.2: Time Area Diagram Ordinates for Kentucky River Watershed

Time (Days)	TAC Diagram Ordinates (a_i) (Km^2)	TA Diagram Ordinates (A_i) (Km^2)
0	0.0	0.0
1	3942.14	3942.14
2	6301.26	10243.40

4.3.1.2 Non-linear Reservoir Component

The attenuation effects in a watershed are normally accounted by routing the inflows of the watershed through a reservoir. In this study, the watershed hypothesized as a non-linear reservoir. This is similar to the concept proposed by Sugawara (1961). The development of the flow routing equation for a non-linear reservoir is described in the following paragraphs.

Let us consider the discharge of water from a non-linear water tank as shown in Figure 4.5. Since the velocity of the water surface in the water tank is very small, the velocity at the outlet at the bottom of the water tank can be given by the following equation:

$$u = \sqrt{2gH} \tag{4.24}$$

where u is the flow velocity at the outlet (m/sec), g is the gravitational acceleration (m/sec^2), and H is the water depth in the tank (m). Using the continuity equation between the water surface and the outlet of the water tank, we get

$$-\bar{A}\frac{dH}{dt} = C_d a u \tag{4.25}$$

where $\bar{A}$ is the horizontal cross-sectional area of the water tank, C_d is a discharge coefficient of the outlet, a is the cross-sectional area of the outlet, and t is the time.

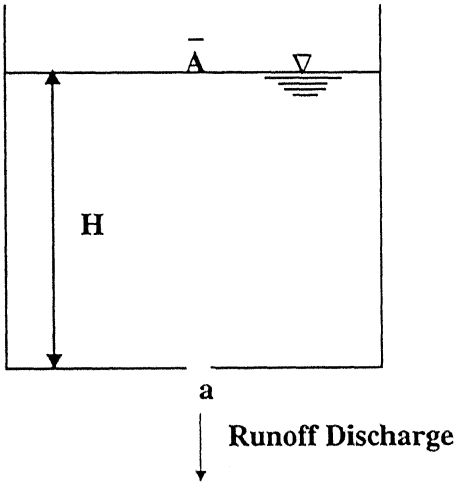


Figure 4.5: A Non-linear Water Tank

Defining

$$h = C_d^2 a^2 H; \quad A = \frac{\bar{A}}{C_d^2 a^2} \tag{4.26}$$

Equation 4.25 becomes

$$-A\frac{dh}{dt} = \sqrt{2gh} \tag{4.27}$$

Thus, h and A are defined as the modified water depth and the modified cross-sectional area, respectively. In order to obtain a solution to equation 4.27, the functional form of A, must be a function of h, as follows (Mizumura 1995):

$$A = f(h) = \frac{T}{\sqrt{h}} \tag{4.28}$$

where T is a constant coefficient. Substituting equation 4.28 in to equation 4.27, the water depth h can be solved as:

$$h = k_1 \exp\left(-\frac{\sqrt{2g}}{T}t\right) \tag{4.29}$$

where k_1 is a constant coefficient. If the water tank represents a catchment in the field, the discharge from the water tank during recession is given by following equation:

$$Q = Q_o \exp(-K_1 t) \tag{4.30}$$

where Q_o is the starting flow at $t = 0$, and K_1 is the recession coefficient. Considering $Q = C_d a u = \sqrt{2gh}$ we obtain:

$$h = \frac{Q_o^2}{2g} \exp(-2K_1 t) \tag{4.31}$$

Comparing the equations 4.31 and 4.29 would give the following relationship:

$$2K_1 = \frac{\sqrt{2g}}{T} \tag{4.32}$$

Now, in order to determine routing equation for the non-linear tank represented by the watershed, let us consider a time step Δt between the times t and $t+\Delta t$. Let the average input to the watershed during Δt be the average effective rainfall represented by $\overline{ER}$, and the average output during Δt be the rate of outflow $\overline{Q}$. According to the continuity equation, the net input to the watershed (non-linear tank) should be equal to the change in storage of the watershed during Δt. This can be mathematically represented by the following equation:

$$(\overline{ER} - \overline{Q})\Delta t = \int_{h_1}^{h_2} S \, dh \tag{4.33}$$

where h_1 and h_2 are the depths in the water tank at times t and $t+\Delta t$, respectively, and dh is the change in water depth during time Δt ($dh = h_1 - h_2$)

Using equation 4.27, and 4.28, equation 4.33 can be transformed into:

$$\left(\overline{ER} - \frac{\sqrt{2gh_1} + \sqrt{2gh}_2}{2} \right) \Delta t = 2T(\sqrt{h_2} - \sqrt{h_1}) \qquad (4.34)$$

Therefore, water depth at time step $t + \Delta t$ is given by:

$$h_2 = \left(\frac{\overline{ER}\,\Delta t + 2T\sqrt{h_1} - \Delta t \sqrt{gh_1/2}}{2T + \Delta t \sqrt{g/2}} \right)^2 \qquad (4.35)$$

Using equation 4.34 and $Q = \sqrt{2gh}$, the flow Q_1 can be given by following equation:

$$Q_2 = \sqrt{2gh_2} = \frac{2K_1\,\overline{ER}\,\Delta t + Q_1\,(2 - K_1\,\Delta t)}{2 + K_1\,\Delta t} \qquad (4.36)$$

Taking the effective rainfall ($\overline{ER}$) at time step $t+\Delta t$ to be the average effective rainfall at time steps $t+\Delta t$ and t and Δt=1 day, the following equation can be obtained for the surface flow component at time step $t+\Delta t$.

$$QS_{t+\Delta t} = C_1\left(\frac{ER_t + ER_{t+\Delta t}}{2}\right) + C_2\, QS_t \tag{4.37}$$

Therefore the following equation can be written to compute the surface flow component at time step t (QS_t):

$$QS_t = C_1\left(\frac{ER^*_{t-1} + ER_t}{2}\right) + C_2\left(QS^*_{t-1}\right) \tag{4.38}$$

where

$$C_1 = \frac{2K_1}{2+K_1} \tag{4.39}$$

$$C_2 = \frac{2-K_1}{2+K_1} \tag{4.40}$$

where, ER^*_{t-1} is the updated amount of effective rainfall at a time step t-1, ER_t is the effective rainfall at a time step t, C_1 and C_2 are the non linear reservoir routing coefficients, QS^*_{t-1} is the updated surface flow component at time step t-1, and K_1 is the

recession coefficient to be determined. The routing coefficients were computed adaptively using observed stream flow on previous two days and calculating the recession coefficient (K_1) as follows (Singh, 1988):

$$K_1 = \log_e\left(\frac{QO_{t-2}}{QO_{t-1}}\right) \tag{4.41}$$

The routing coefficients (C_1 and C_2) were computed using equations 4.39 and 4.40 respectively. The overall model structure of the CRR Model – II can be summarized in the following steps:

1. Compute the daily average rainfall using arithmetic mean method, and actual daily evapotranspiration following Haan's procedure.
2. Compute the base flow using equation 4.1
3. Compute the non-linear reservoir routing coefficients C_1 and C_2 (equations 4.39 through 4.41) using observed streamflow data from previous time steps.
4. Compute the initial available soil moisture using equation 4.10 and then compute initial soil moisture storage.
5. Compute the potential incremental infiltration amount at a time step t (ΔF_t) using the equation 4.6, actual incremental infiltration at time t (ΔFA_t) by comparing ΔF_t with P_t, and then the effective rainfall using equation 4.3.
6. Run the effective rainfall amounts through the time area concentration diagram and compute the convoluted effective rainfalls to be used in the non-linear tank model as inflow.

7. If the streamflow is on the rising limb then compute the updated inflow component (ER^{*}_{t-1}) using the equations 4.21, and compute the updated surface flow component (QS^{*}_{t-1}) using equation 4.20. Then compute surface flow component (QS_t) using equation 4.38 and add it to base flow to predict the streamflow.
8. If the streamflow is on the falling limb then use decay model (equation 4.22 and 4.23) to predict the streamflow.
9. Update all parameters and stores using appropriate equations described above.

4.4 Study Area and Data

The data derived from the Kentucky River basin, Kentucky USA were employed to calibrate and validate all the models developed in this research. The Kentucky River basin, shown in Figure 4.6, encompasses over 4.4 million acres of the state of Kentucky, USA. Forty separate counties lie either completely or partially within the boundaries of the river basin. The Kentucky River is the sole water supply source for several water supply companies of the state. There is a series of fourteen Locks and Dams on the Kentucky River, which are owned and operated by the US Army Corps of Engineers. The drainage area of the Kentucky River at Lock and Dam 10 (LD10) near Winchester, Kentucky is approximately 6,300 km^2. The data used this research include average daily streamflow (m^3/s) from Kentucky River at LD10, and daily total rainfalls (mm) from five rain gauges, Manchester, Hyden, Jackson, Heidelberg, and Lexington Airport scattered throughout the Kentucky River Basin (see Figure 4.6). The total length of the rainfall-runoff data of 30-years (1960-1989) was available. The data were divided into two sets: a calibration data set consisting of daily rainfall and flow data for thirteen years (1960-1972), and a validation data set of thirteen years (1977-1989). The data for

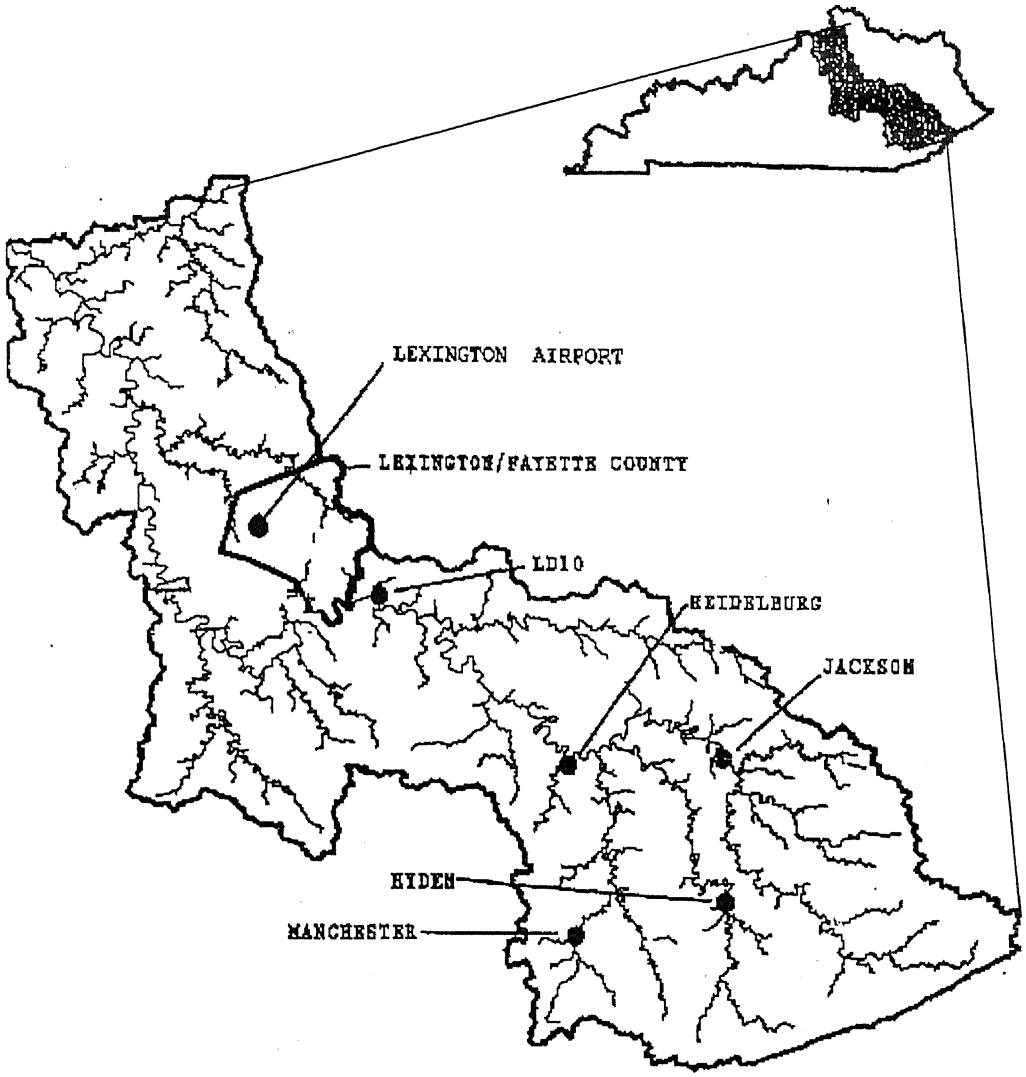


Figure 4.6: Kentucky River Basin

the period 1973-1976 were excluded from model development due to many missing records during that period.

4.5 Calibration of CRR Models

Once the model structure has been identified, the next step in the model development is the calibration of the model. The CRR models described above consist of a total of nine parameters. These are soil moisture accounting model parameters β_0 and β_1; Green-Ampt infiltration equation parameters K, ψ, η, and S; and the reservoir routing coefficients K_1, C_1, and C_2. In the CRR Model–I, the linear reservoir routing coefficients are constant; whereas, these were determined adaptively using past streamflow data in the CRR Model-II. The values of the SMA model's regression coefficients β_0 and β_1, were determined by first finding initial available soil moisture storages for some representative storms taken from the Kentucky River basin using the calibration capability of the computer flood hydrograph package HEC-1 (HEC-1, 1990), and then regressing the initial available soil moisture storages at the beginning of a storm against the corresponding base flows. The regression coefficients were determined to be β_0 = 1.158832, and β_1 = -0.000107, with correlation coefficient of 0.8410.

The calibration of infiltration parameters was performed using two different approaches. The first approach used HEC-1 flood hydrograph package and the second method used the *elitist real-coded* GA (*elitist* RGA). The HEC-1 flood hydrograph package (HEC – 1, 1990) provides the capability to calibrate the infiltration parameters using historical rainfall and flow data. The data from some representative storms taken from Kentucky River watershed was employed to determine the Green-Ampt infiltration parameters.

However, the HEC-1 uses rainfall and flow data for certain number of storms only. The *real-coded* GA allows the use of continuous data for the determination of model parameters, and was therefore employed. The description of *real-coded* GA, *elitism*, its parameters, and its importance was described in Chapter 3. The efficiency and effectiveness of the developed *elitist real-coded* GA program was evaluated on standard mathematical functions prior to applying it for the CRR model calibration. The performance of an optimization algorithm can be measured by its robustness and efficiency. The robustness is interpreted as the probability of finding the global optimum from a series of independent trials. The efficiency is determined by number of functional evaluations required by the algorithm to satisfy prescribed convergence criteria.

4.5.1 Validation of *Elitist Real- Coded* GA

A set of standard test functions was chosen from literature to test effectiveness and efficiency of a simple *real-coded* genetic algorithm (RGA) and *elitist real-coded* GA (*elitist* RGA). Four different mathematical functions, namely, Rastrigin, Six-Hump Camelback, Hartman, and Griewank functions were selected for testing RGA and *elitist* RGA in the present study based on two factors: dimensionality of the functions, and the number of local optima on their error surface. Further, these function optimization problems using simple RGA and *elitist* RGA are compared with the shuffled complex evolutionary algorithms SCE1 and SCE2 (Duan et al., 1993) and with the improved genetic algorithm (IGA) developed by Ndiritu and Daniell (2001). The SCE algorithm (Duan et al., 1993) is a combination of random and deterministic approaches, the concepts of clustering, systematic evolution of complex of points, and competitive evolution. The IGA is a combination of fine-tuning, hill-climbing and independent sub-

population search with shuffling methodologies. The details of mathematical functions used for optimization are given in Appendix-A.

The *elitist* RGA and the simple RGA were tested by running 100 trials on each test function. All the test functions were arranged such that they have their optimum value at zero. An optimization run was considered a success, if the function value reduced to at least 0.001. Further, the optimization run was considered a failure if it reached 25,000 function evaluations, or if the region spanned by the population converged to within 10^{-12} of the parameter range in each direction without the 0.001 mark being attained. Using RGA, the experimentation was carried out by maintaining the balance between population size and the number of generations to fix the maximum function evaluations. The RGA with and without *elitism* was tested on all the test problems by keeping the maximum function evaluations at 25000. The values of the distributive indices η_c and η_m of 0.02 and 2, were used for the crossover and mutation, respectively, in this study. Further, the crossover (P_c) and mutation (P_m) probabilities were fixed at 1.0 and 0.05 for comparison purpose with the IGA developed by Ndiritu and Daniell (2001). The tournament selection with tournament size of two was considered in all the test problems for optimization using both RGA and *elitist* RGA. The population size was fixed based on the trials to get better efficiency of the algorithm.

Rastrigin Function

The dimensionality of the Rastrigin function is two and the function has more than 50 local minima in the region of interest. The global minimum is 0 at a point (0,0). The optimum population size considered for this function is 40. Figure 4.7 shows the performance of RGA and *elitist* RGA with the multi start simplex (MSX), controlled

random search (CRS2), and shuffled complex evolutionary algorithms SCE1 and SCE2 (Duan et al., 1993) in optimizing Rastrigin function. It can be observed from Figure 4.7 that the RGA performs comparable to the MSX algorithm in terms of successes rate but poorer in terms of maximum function evaluations for 100% success with CRS2, SCE1 and SCE2 algorithms. The minimum number of function evaluations (752) was obtained from SCE2 algorithm with 99% success. However, *elitist* RGA gives optimum value at an average of 228 (Figure 4.7) function evaluations with 100% success, and is deemed to the best.

Six Hump Camel Back Function

Figure 4.8 shows the comparative performance of various algorithms in optimizing Six Hump Camel Back function. This function has dimensionality of two and the global minimum is 0 at (0.08983, -0.7126) and (-0.08983, 0.7126). The function is symmetric about origin and has three pairs of local minima. The population size considered for the optimal runs is 10. The performance of RGA is relatively poorer as compared to MSX, CRS2, SCE1 and SCE2 but the *elitist* RGA performs equally comparable with other algorithms.

Hartman Function

The comparative study for Hartman function of RGA and *elitist* RGA with the MSX, CRS2, SCE1, SCE2(Duan et al., 1993), and IGA (Ndiritu and Daniell, 2001) is shown in Figure 4.9. The Hartman function has the dimensionality of six and has four local minima in the region of interest. The global minimum is 0 at a point (0.201, 0.150, 0.477, 0.275, 0.311, 0.657). The optimum population size for this function optimization was found to be 80. The RGA performs better compared to SCE1, SCE2 and CRS2 with

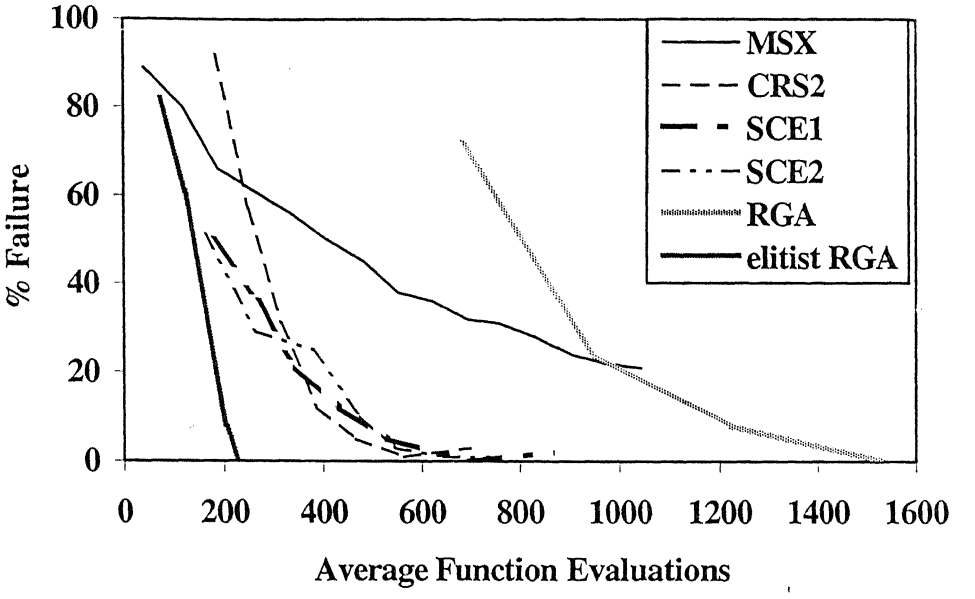


Figure 4.7: Comparative Performance for Rastrigin Function

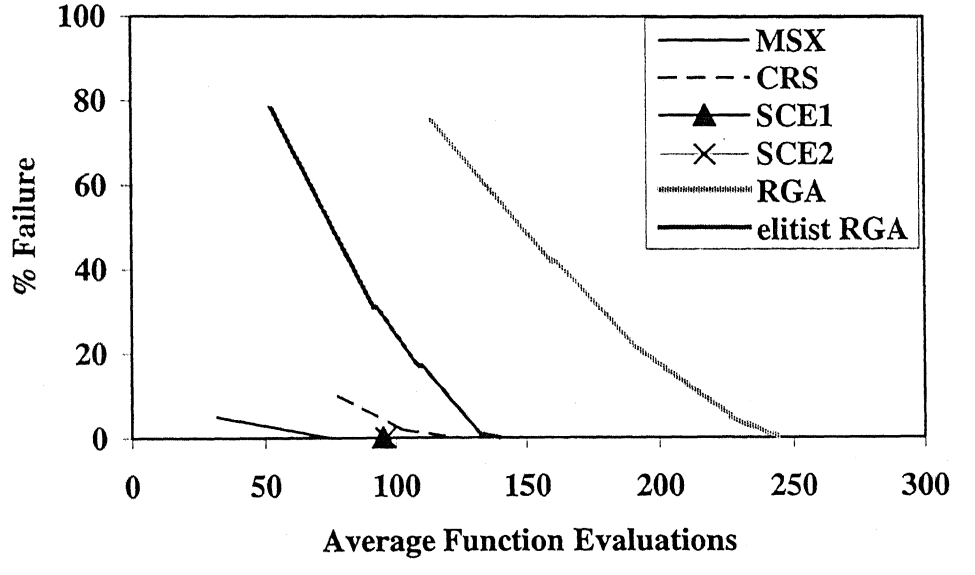


Figure 4.8: Comparative Performance for Six Hump Camelback Function

90% success with 2,913 functional evaluations and comparable with MSX and IGA in terms of the number of the function evaluations. The *elitist* RGA optimizes the function with less number of function evaluations at 2,560 but with a success rate of 88%. The simple GA achieved 11% success for the same function (Ndiritu and Daniell, 2001).

Griewank Function

The Griewank function has the dimensionality of 10 and the global optimum of $x_i = 0$, (i = 1,...., 10). The function possesses thousands of local minima. The optimum population size for this function was found to be 26. The RGA achieved with 100% success (Figure 4.10) with an average number of 6,203 function evaluations. On the other hand, *elitist* RGA achieved 100% success rate with an average function evaluations of 3,811 only in optimizing the Griewank function. *Elitist* RGA was found to be far better than the parallel GA used by Mühlenbein et al. (1991) with average functional evaluations 59,520, and IGA used by Ndiritu and Daniell (2001) with average function evaluations of 101,096 in optimizing Griewank function. Further *elitist* RGA was equally compatible with the average function evaluation of 3,070 from SCE2 algorithm. The simple GA fails in all trials when applied to Griewank function optimization (Ndiritu and Daniell, 2001; Tomassini, 1993).

In a nutshell, it can be noted that the *elitist* RGA performs better than most of the existing algorithms, and comparable with the remaining ones. Further, Franchini (1996) indicated that high precision is required in the search for the global minimum of the objective function. The inherent advantage of RGA is that it works well where the problems need high precision because of dealing with the floating-point representation of the variables. RGA is based on the working principles of simple GA with the

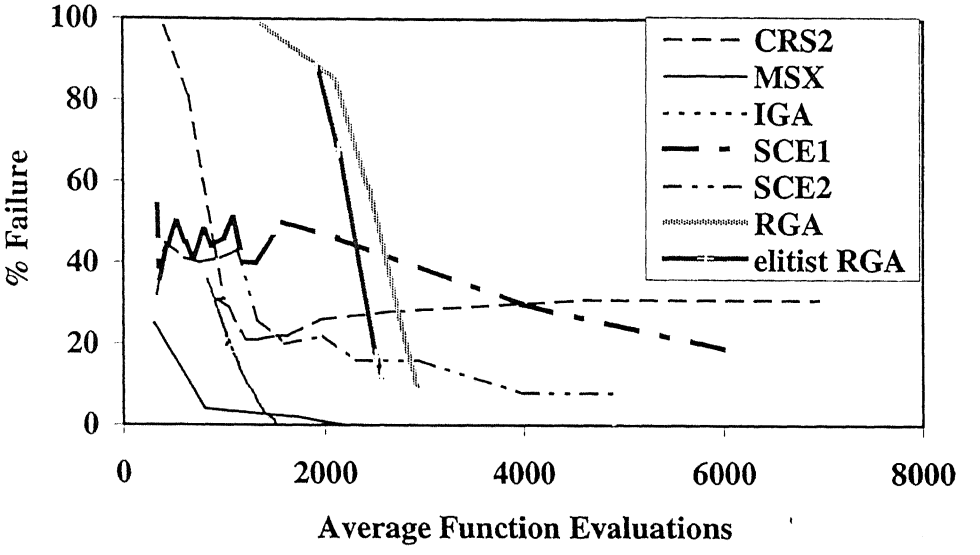


Figure 4.9: Comparative Performance for Hartman Function

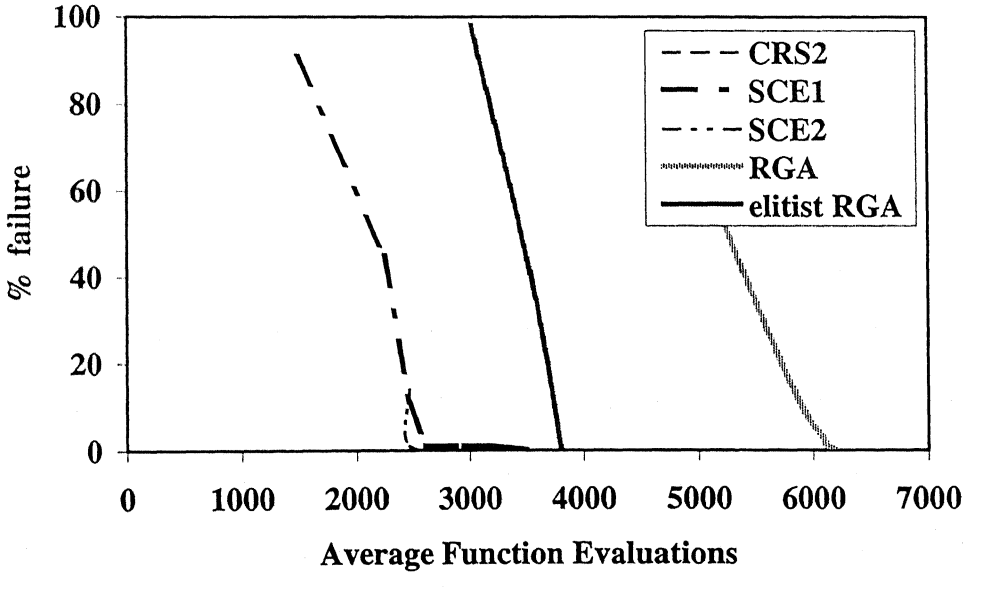


Figure 4.10: Comparative Performance for Griewank Function

operators designed to work on real-parametric space. Further, with the addition of *elitism*, the *elitist* RGA performed consistently well with all the four test problems. It is equally compatible with IGA in optimising Hartman function, and better with SCE algorithms. Further, the *elitist* RGA superior in optimising Griewank function in comparison with IGA and parallel GA and equally comparable with SCE algorithm. Further, the IGA and SCE algorithm have more complex in nature as compared to *elitist* RGA. The relatively poor efficiency of the IGA in optimising the Griewank function in comparison with the optimising the Hartman function can be attributed to the number of local optima in the two problems. The Hartman function possesses four optima, while the Griewank function possesses greater than a thousand optima in the region of interest. The *elitist real-coded* GA and SCE performed superiorly compared to IGA in optimising Griewank function. The success of *elitist* RGA in optimising all the functions indicates that the use of real parameters in the continuous search space using simple evolutionary approach with elitism is better approach than the other complex algorithms.

4.5.2 Calibration of Green-Ampt Parameters

The calibration of an infiltration model makes use of known rainfall and runoff data, and involves minimization of errors between the observed and estimated runoff. The runoff was calculated using the two CRR models developed in this study. In solving an optimization problem, an objective function needs to be formulated. In the context of the present study, the objective function may consist of the sum of squares of the differences between the observed and estimated runoff for the known set of rainfall and runoff data. The objective function employed in the present study can be represented by the following equation:

Minimize

$$E = \frac{1}{N}\sum_{t=1}^{N}(Q(t)-QO(t))^2 \tag{4.42}$$

where E is the error function to be minimized in the optimization formulation, $Q(t)$ is the estimated runoff at time t, $QO(t)$ is the observed runoff at time t, and N is the total number of data points in the observed runoff series used for calibration. The $Q(t)$ was computed using the two CRR models employed in this study. The *elitist* RGA was employed to determine the optimal set of Green-Ampt infiltration parameters using the objective function represented by equation (4.42) above. The RGA parameters $\eta_c = 2$, $\eta_m = 20$, $P_c = 0.9$, and $P_m = 0.01$ and population size of 40, were used. The Green-Ampt infiltration parameters were calibrated using 13 years of calibration data set for both the CRR models. The calibrated values of Green-Ampt parameters from the two CRR models are given in Table 4.3.

The Green-Ampt infiltration parameters obtained from the two methodologies for two CRR model are presented in Table 4.3. It can be noted from Table 4.3 that the infiltration parameters obtained using the technique of *elitist* RGA from both the CRR models match exceedingly well. Further, the parameters compare reasonably well with those obtained using the standard HEC-1 model. Once the calibration of CRR models is complete, the corresponding parameters were used to validate the two models using 13 years testing data set (1976-1989) by computing various statistical performance indices.

Table 4.3: Green - Ampt Infiltration Parameters

Parameter	HEC-1	RGA Calibrated Parameters	
		CRR Model - I	CRR Model -II
Soil Suction Head, ψ (mm)	221.742	201.267	201.021
Hydraulic Conductivity, K (mm/hr)	0.254	0.1999	0.1999
Porosity, η	0.100	0.09686	0.11229
Maximum Depth of Soil Storage, S (mm)	304.8	305.42	305.66

4.5.3 Performance Evaluation Indices

In order to properly evaluate the model performance, it is important to clearly define the criteria by which the performance of the model will be judged. Most of the studies on rainfall-runoff modeling have employed performance statistics such as mean square error (MSE), root mean square error (RMSE), etc. in order to evaluate the efficiency of the models and to assess the effectiveness of the model in prediction. Such statistics have forms that are similar to the objective function in nature. Moreover, such error statistics can be biased towards high magnitude flows because of their formulation.

Dawson and Wilby (2001) noticed that performance statistics based on the squared errors do provide a general measure of model performance, but they do not identify specific regions where a model is deficient. Therefore, it will be desirable to consider

certain other statistical measures that are unbiased, and have different form in order to test the effectiveness of the developed models in terms of their prediction ability. Eight different standard performance evaluation indices were used in this study to evaluate the relative strengths and weaknesses of the various models developed. These are threshold statistics (TS), average absolute relative error (AARE), coefficient of correlation (R), efficiency (E), normalized mean bias error (NMBE), normalized root mean square error (NRMSE), percentage deviation from maximum flow (%MF), and the persistence factor (E_{per}). The AARE gives average prediction error without any bias; whereas the threshold statistics gives the distribution of errors. The NMBE statistic indicates over-estimation or under-estimation in the computed values of the physical variable being modeled, and provides information on long-term performance (Stone, 1993; and Safi et al., 2002). The NRMSE statistics provides information for short-term performance of a model by allowing term-by-term comparison of the actual differences between the estimated and the observed values. The persistence factor (E_{per}) can assess the effectiveness of the developed model over a simple persistence model (Porporato and Ridolfi, 2000). Further the other performance evaluation indices (R, E, etc.) assess the efficiency of the models. The combination of these performance evaluation indices is needed to give an idea about the model ability in terms of both efficiency and effectiveness. In their earlier study, Dawson and Wilby (1998) recommended to use; the mean square relative error (MSRE) as unbiased estimation criteria for improvement in training of low flows. Further, Kumar and Minocha (2001) suggested the relative error criteria to assess the model performance. The TS and AARE statistics, which are relative in nature, have been used extensively in the past (Jain et al., 2001; Jain and Ormsbee, 2002; and Jain and Indurthy, 2003). The performance of each of the model

developed in this study was measured in terms of these eight different standard performance evaluation indices described as follows:

1) Average Absolute Relative Error (AARE)

The Average Absolute Relative Error (AARE) is the average of the absolute values of the relative errors in forecasting certain number of data points. To compute *AARE*, first relative error in forecasting needs to be calculated. The relative error is a measure of the error in forecasting a particular variable relative to its exact value. Mathematically, *AARE* can be computed using the following equations:

$$RE(t) = \frac{Q(t) - QO(t)}{QO(t)} \times 100\% \tag{4.43}$$

$$AARE = \frac{1}{N} \sum ABS[RE(t)] \tag{4.44}$$

Where $QO(t)$ is the observed flow value at time t, $Q(t)$ is the predicted flow value at time t, $RE(t)$ is the relative error in forecasting flow at time t, *AARE* is the average absolute relative error, N is the total number of data points forecasted, and the summation runs from 1 to N. As obvious from its definition, lower *AARE* values will indicate better effectiveness of model prediction and vice-versa.

2) Threshold Statistics (TS_x)

The threshold statistic gives the distribution of the errors that enables one to get a clearer picture of the model performance. The threshold statistic is always defined for a certain level of absolute relative error (ARE), say $x\%$ and is designated by TS_x. The TS_x may be defined as the percentage of data points forecasted for which the *ARE* is less than $x\%$. Mathematically, it can be calculated as follows:

$$TS_x = \frac{n}{N} \times 100\% \tag{4.45}$$

Where n is the number of data points forecasted whose ARE is less than $x\%$ and N is the total number of data points forecasted. Threshold statistics were computed for ARE levels of 5%, 10%, 25%, 50% and 100% in this study. As obvious from its definition, higher the value of threshold statistic better is the model performance, and vice-versa.

3) Correlation Coefficient (R)

The correlation coefficient measures the strength of correlation between the modeled output and observed output. Its value ranges between –1 & +1. The value close to 1.0 indicates good model performance, and a value close to 0.0 means poor model performance. The coefficient of correlation can be calculated using the following equation:

$$R = \frac{\sum_{t=1}^{N}\left(QO(t)-\overline{QO}\right)\left(Q(t)-\overline{Q}\right)}{\sqrt{\sum_{t=1}^{N}\left(QO(t)-\overline{QO}\right)^2\left(Q(t)-\overline{Q}\right)^2}} \tag{4.46}$$

Sometimes, higher value of R may not necessarily indicate better performance of the model because of the tendency of the model to be biased towards higher or lower values (Ehrman *et al.*, 2000). There should be another measure to ascertain the strength of the correlation between observed and modeled outputs, which is coefficient of efficiency.

4) Coefficient of efficiency (E)

The coefficient of the efficiency (Nash-Sutcliffe, 1970) compares the modeled and observed values of the variable and evaluates how far the model is able to explain total variance in the data set. The coefficient of efficiency can be calculated using the following equations:

$$E = \frac{E_1 - E_2}{E_1} \tag{4.47}$$

$$E_1 = \sum_{t=1}^{N} (QO(t) - \overline{QO})^2$$
$$E_2 = \sum_{t=1}^{N} (Q(t) - QO(t))^2 \tag{4.48}$$

where $\overline{QO}$ is the mean of observed values, and other variables are same as explained earlier. As obvious, higher the value of efficiency from a model, better is the performance. According to Shamseldin (1997) the efficiency (E) above 90% indicates very satisfactory performance, a value in the range of 80-90% indicates fairly good performance, and a value below 80% indicates an unsatisfactory fit.

5) Normalized Mean Bias Error (NMBE)

The Normalized Mean Bias Error (NMBE) can be computed using the following equation:

$$NMBE = \frac{\frac{1}{N}\sum_{t=1}^{N}(Q(t) - QO(t))}{\frac{1}{N}\sum_{t=1}^{N} QO(t)} X\ 100\ \% \tag{4.49}$$

It is obvious from its definition, positive value of NMBE would indicate an overall over-prediction while negative value of NMBE would mean an overall under-prediction from the model.

6) Normalized Root Mean Square Error (NRMSE)

The Normalized Root Mean Square Error (NRMSE) can be computed using the following equation:

$$NRMSE = \frac{\left(\frac{1}{N}\sum_{t=1}^{N}(Q(t)-QO(t))^2\right)^{1/2}}{\frac{1}{N}\sum_{t=1}^{N}QO(t)} \qquad (4.50)$$

It is to be noted that NRMSE statistics is relative with respect to the total sum of observed flow, however, it can still be biased towards high magnitude flows. The AARE is relative with respect to individual flow values is better indicator as it will not be biased towards either high or low magnitude flows. Further, it is obvious from its definition that lower value of NRMSE would indicate better model performance and vice-versa.

7) The Relative Error in the Maximum Flow (% MF)

The relative error in the maximum flow (*%MF*) provides information about the percentage of over- estimation or under-estimation in predicting the maximum flow. This can be computed using the following equation.

$$\%MF = \frac{Q_{max} - QO_{max}}{QO_{max}} X\,100 \qquad (4.51)$$

Where Q_{max} is the modeled value of maximum flow (m^3/sec), and QO_{max} is the observed value of maximum flow (m^3/sec). Clearly, lower value of $\%MF$ means better model performance vice-versa.

8) Coefficient of Persistence (E_{pee})

The coefficient of persistence (E_{per}) is commonly used to evaluate quality of the forecast. It gives an idea about how good a model in performing in comparison to a persistence model, where predicted flow at time t is taken equal to the observed flow at a previous time (t-1). The value of the coefficient of persistence (E_{per}) can be calculated using the following equations:

$$E_{per} = \frac{E_1 - E_2}{E_1} \tag{4.52}$$

$$E_1 = \sum_{t=1}^{N} \left(QO(t) - QO(t-1)\right)^2$$

$$E_2 = \sum_{t=1}^{N} \left(Q(t) - QO(t)\right)^2$$

The values of the coefficient of persistence larger than zero ensure that the forecast is better than that obtained from a simple persistence model (Porporato and Ridolfi, 2000). As obvious, higher the value of persistence from a model better is the performance.

4.6 Discussion of Results from CRR Models

Once the calibration of the two CRR models was complete, the two model structures were used to compute daily streamflow in Kentucky River at LD10 during both calibration and validation periods. Then, various performance evaluation indices were

computed during both calibration and validation periods of 13-years each. The computed values of the performance evaluation indices are presented in Table 4.4.

Table 4.4 presents the values of threshold statistics for absolute relative error (ARE) levels of 1-, 5-, 25-, 50-, and 100-percent, AARE, R, E, NMBE(%), NRMSE, %MF, and E_{per} computed from the two CRR models during both calibration and validation data sets. According to Shamseldin (1997) criteria, both CRR models developed in this study have *E* more than 80%, and the performance can be characterized as fairly good both during calibration and validation (see Table 4.4). It can be observed from Table 4.4 that the model that uses combination of linear channel and non-linear reservoir with adaptive routing coefficients (CRR Model-II) performs better than the CRR Model–I, during both calibration and validation periods. As expected, the CRR Model-II obtained better AARE of 21.29% and the best threshold statistics up to 50% ARE level during calibration. It is clear from Table 4.4 that the CRR Model-II consistently outperformed the CRR Model–I as in terms of R, E, and NRMSE, %NMBE and E_{per} statistics, through %MF statistics was slightly better from CRR Model -I during calibration.

Looking at the results obtained during validation data set of 13 years (1977-1989) in Table 4.4, it can be observed that the CRR Model–II again performed far better than CRR Model–I in terms of all threshold statistics, AARE, NMBE, R, E, NRMSE, and E_{per} and comparable to CRR Model-I in terms of %MF statistics. The CRR Model-II was able to achieve the minimum AARE of 22.95% and the minimum NMBE value of 0.205% and attains maximum persistent factor (E_{per}) of 0.372. The value of persistence factor above zero shows the effectiveness of conceptual models over simple persistence model. Also, the value E_{per} from CRR Model-II is more in magnitude over that from

Table 4.4: Performance Statistics from CRR Models

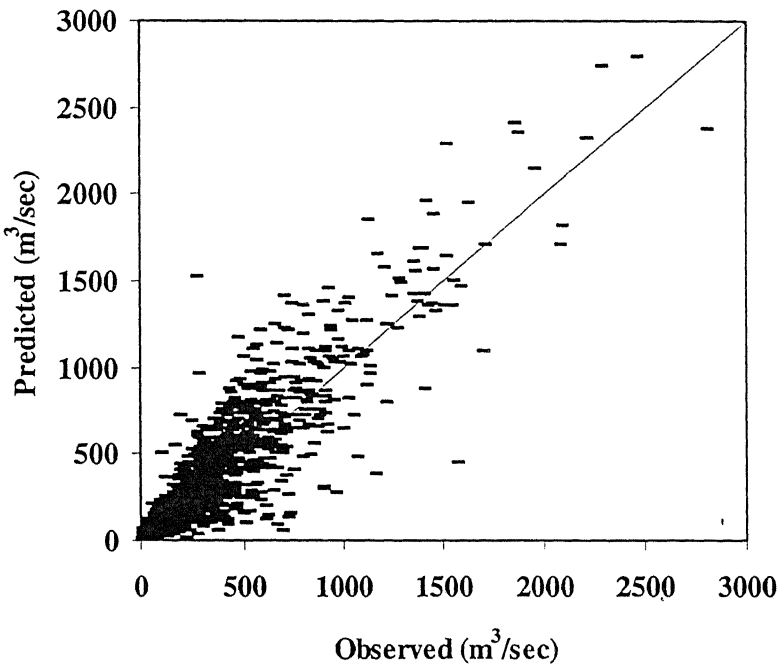
Model	TS1	TS5	TS25	TS50	TS100	AARE	R	E	NMBE(%)	NRMSE	%MF	E_{per}
During Calibration												
CRR Model-I	6.10	18.87	68.52	89.26	97.81	23.57	0.9363	0.8436	8.951	0.639	-3.39	0.109
CRR Model-II	7.64	30.49	77.65	91.57	97.49	21.29	0.9452	0.8845	0.188	0.553	-5.30	0.319
During Validation												
CRR Model-I	4.84	18.32	67.89	88.66	97.59	24.68	0.9332	0.8344	9.462	0.649	-15.25	0.123
CRR Model-II	8.78	34.08	77.94	90.73	97.34	22.95	0.9438	0.8814	0.205	0.550	-20.12	0.372

Model-I. This shows that the CRR Model-II has better predicting capability over the CRR Model-I.

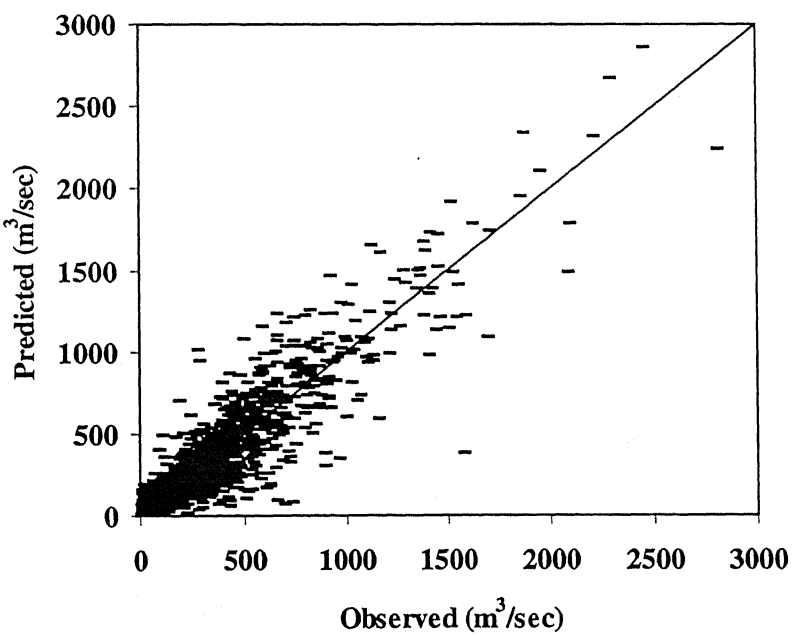
The results in graphical form are presented in Figure 4.11 and Figure 4.12. Figure 4.11 show the observed and predicted flow values (m^3/sec) against each other as a scatter plots for the validation data set of 13-years (1977-1989) from the CRR Model–I and II. There seems to be a systematic error in predicted flows from the two CRR models as the predicted flows tend to deviate from the 45^0 line. This trend is more pronounced from CRR model-I indicating superiority of CRR Model-II in modeling the complex, dynamic, and non-linear rainfall-runoff process. Figure 4.12 shows the observed and predicted flow values against time (in days) for the dry and wet years of 1986 and 1989 from CRR Model-II. It can be noticed from Figure 4.12 that the predicted flow values obtained from Model-II match reasonably will with the observed flow values except for some higher magnitude flows. From the above analysis it is taken to that the CRR Model -II has chosen as better model for Kentucky River basin.

4.7 Summary

This chapter describes the development of the two conceptual rainfall-runoff (CRR) models, both CRR models assume the hydrologic system to be made up of two storage components, a surface store and a sub-surface store. The two CRR models are similar in structures in modeling the base flow, infiltration, soil moisture accounting, and evapotranspiration but differ in the manner in which the surface flow component is modeled. The CRR Model-I accounts for both translation and attenuation of the watershed by modeling the surface flow component by assuming the watershed to be a

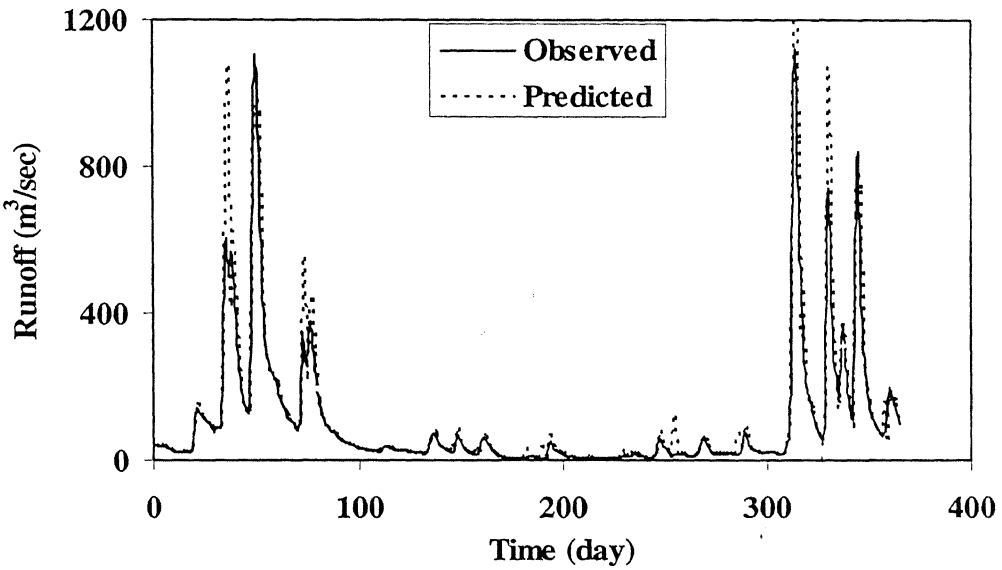


(a) From CRR Model -I

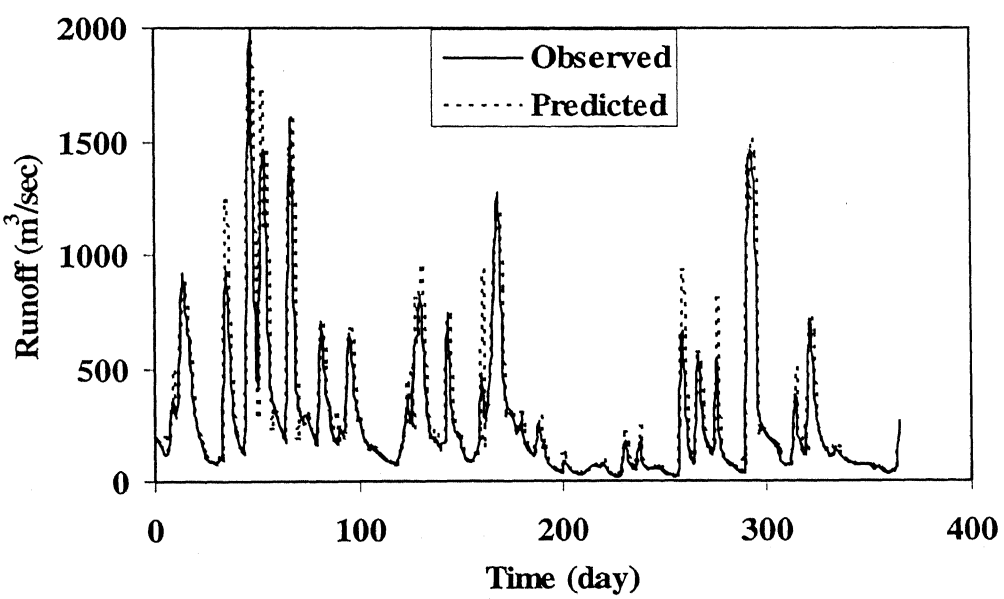


(b) From CRR Model –II

Figure 4.11: Observed and Predicted Flows during Validation Period



(a) From the Year 1986



(b) From the Year 1989

Figure 4.12: Observed and Predicted Flows from CRR model -II

linear reservoir. The CRR Model-II uses a linear channel element in the form of a time area diagram to account for the translation effects and accounts for the attenuation effects by routing the output from the linear channel element from a non-linear reservoir representing the watershed. The Green-Ampt infiltration equations were employed to compute infiltration, as they provide exact analytical solution to an approximate physically based model, and also allow the use of a mechanism to continuously update the soil moisture storage. The daily rainfall and streamflow data derived from the Kentucky River watershed, USA were employed to calibrate and validate the two CRR models. The calibration of both the CRR models was performed using both HEC-1 and the *elitist real-coded* genetic algorithm (*elitist* RGA). The *elitist* RGA source code was first validated by optimizing four mathematical functions, Rastrigian function, Six-Hump Camel Back function, Hartman function, and the Griewank function, to compare the results with the best results available in literature. Eight different performance evaluation indices were computed from the two CRR models both during calibration and validation data sets in order to evaluate the relative strengths and weaknesses of the two CRR models.

The values of the Green Ampt infiltration parameters (see Table 4.3) obtained using the *elitist* RGA from both the CRR models matched exceedingly well. The *elitist* RGA determined infiltration parameters matched reasonably well with the HEC-1 determined infiltration parameters. It was found that the porosity and the maximum depth of soil storage from HEC-1 and *elitist* RGA were very close but the soil suction head and the hydraulic conductivity values differed from the two methodologies. The *elitist* RGA determined parameters were finally selected to be used to compute the streamflow for validation of the two models. The results in terms of various performance evaluation

indices from the two CRR models are presented in Table 4.4. The performance of both the CRR models in terms of various performance evaluation indices was found to be reasonably good. However, as expected, the CRR Model-II was found to be superior to the CRR Model-I in modeling the complex, dynamic, and non-linear rainfall-runoff process in the Kentucky River watershed.

Chapter 5

Development of ANN Rainfall-Runoff Models

5.1 General

Artificial Neural Networks (ANNs) have been applied for various purposes such as modeling and prediction, generalization, function approximation, classification and pattern recognition, interpolation, conditional simulation, and system control, etc. The versatility of ANNs coupled with the availability of high speed computers has led to the employment of ANNs as efficient tools of modeling and forecasting engineering systems including hydrologic process modeling. Due to their universal function approximation property, robustness, ability to learn from noisy information, the ANNs are capable of capturing the highly complex, dynamic, non-linear, and fragmented relationships such as in the rainfall-runoff relationship in a watershed.

In this study three different types of rainfall-runoff models have been investigated that employ the technique of ANNs. The first type of ANN models are black box type models that do not consider the underlying physics of the hydrologic process. These are called BANN models. The second type of rainfall-runoff models integrates the ANN and conceptual techniques, and is called integrated models. The integrated models use conceptual techniques to model individual component(s) of the overall hydrologic process, and the output is then embedded into the overall ANN model resulting in the

integrated ANN rainfall-runoff models. Because it is possible to see the details of the physical processes, even in a partial sense, the integrated ANN models can also be termed as *grey-box* models (or GANN models). The first two types of models, i.e. both BANN and GANN models, attempt to model the whole hydrograph using a single ANN or technique. The third type of models investigated in this study decomposes the flow hydrograph into different components corresponding to different dynamics, which are then modeled using different ANN and/or conceptual technique, these are called GDANN models in this study. Each of the three categories of models mentioned above was further divided into two types. The first type of models used popular back-propagation (BP) algorithm for training the ANNs while the second type of models employed *elitist real-coded* genetic algorithm (*elitist* RGA) as the training method. All the ANN models were calibrated and validated using the rainfall and streamflow data taken from the Kentucky River watershed in order to make comparisons with the CRR models developed earlier. This chapter discusses, in detail, the development of different types of ANN rainfall-runoff models.

5.2 Black-Box ANN (BANN) Models

Before applying ANNs for rainfall-runoff modeling, a number of decisions must be made. The development of rainfall-runoff models using ANNs involves number of steps: 1) selection of data set for calibration and validation of model, 2) identification of input vector, 3) normalization (scaling) of the selected data, 4) determination of number of input and output neurons of the ANN model, 5) determining the number of hidden layer neurons 6) training the ANN model and 7) validation of model using selected performance evaluation indices.

The data used (training & testing) for the development of the simple BANN model is similar to the data used (calibration & verification) for the development of conceptual models described earlier. There are multitudes of network types available for ANN applications and the choice depends on the nature of the problem and available data. However, three layer feed-forward network and sigmoid activation function are the most widely used network and transfer function respectively, for rainfall-runoff modeling (ASCE task committee, 2000 a, b; Dawson and Wilby, 2002). The BANN models developed in this study consisted of three layers: an input layer consisting of neurons depending on the input data vector, a hidden layer, and an output layer consisting of neurons depending on the output vector. Further, the sigmoid activation function was used as the transfer function at both hidden and output layers. The sigmoid function is a continuous non-decreasing bounded function in the range 0 to 1. To correlate the output of the network with the bounds of sigmoid function, hydrologic data must be normalized in the range 0 to 1. Because range and units of different variables involved in modeling are different, normalizing the variables and recasting them in dimensionless units helps in developing meaningful relationships. For the present work, the input and output data vectors were normalized in the range 0 to 1 using the following equation:

$$z_{ij} = \frac{x_{ij} - x_{j\min}}{x_{j\max} - x_{j\min}} \tag{5.1}$$

where x_{ij} and z_{ij} are input vector of i^{th} observation of the j^{th} variable before and after normalization, respectively; and x_{jmin} and x_{jmax} are the minimum and maximum values of the j^{th} variable in all observations.

The selection of proper input and output variables and determination of number of hidden neurons essentially decide the architecture of the ANN models. The number of neurons in the hidden layer is, in fact, responsible for capturing (or mapping) the dynamic and complex relationship among various input and output variables considered in developing an ANN. The number of neurons in the hidden layer was determined using a trial and error procedure. In this study, the only neuron in the output layer represented the flow at time t (Q_t). The identification of input vector (neurons) for the BANN models is discussed next.

5.2.1 Input Vector Identification for BANN Models

The input vector for the BANN model selected in this study was based on cross correlation analysis performed between the rainfall in the past for various time steps and flow at time *t* and from partial- and auto-correlation analysis for individual rainfall and flow time series.

Figure 5.1 shows the cross correlation of flow at time *t* with the rainfall lagged in the past. It is showing significant correlation of the flow at time *t* with rainfall up to past four time lags. But the auto-correlation function of rainfall time series, show in figure 5.2, indicates the rainfall values up to two time steps in the past are sufficient because the auto-correlation function first crosses the zero at the lag two. The delay time in the auto-correlation function indicates the temporal persistence of the individual time series (Sivakumar, 2001). Elshorbagy et al. (2002) used the delay time measured from autocorrelation function as the maximum length of missing segment of observations that can be estimated in one step using ANNs. The temporal persistence of individual series shows that the information contained up to lag two is sufficient to explain the

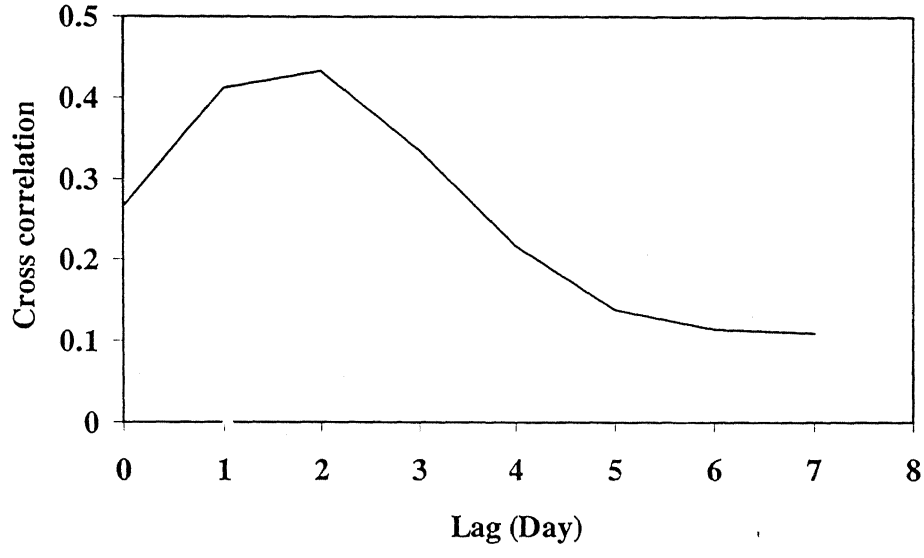


Figure 5.1: Cross Correlation of Rainfall with the Runoff at Time t at Different Lags

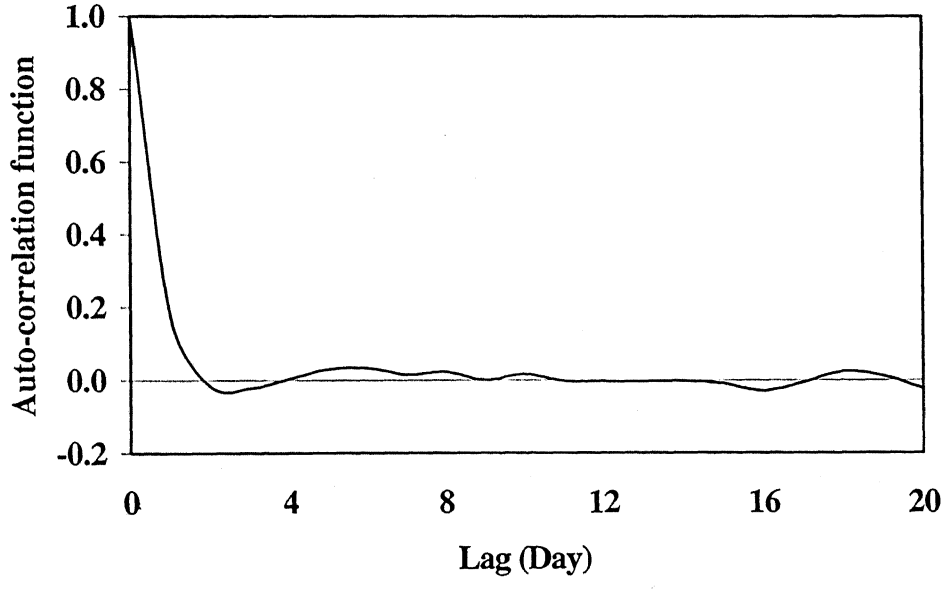


Figure 5.2: Autocorrelation Function of Rainfall Series

dynamics of the rainfall process. The above analysis of cross- and auto-correlation functions suggests that rainfall values up to lag 2 can be considered in the input vector for the BANN model. Further, the time of concentration of the basin is 2 days, which also supports the lag time of the rainfall considered for input vector for the BANN model development.

Figure 5.3 and Figure 5.4 show the auto-correlation and partial auto-correlation function plots of the daily flow time series, respectively. The gradual decaying pattern of the autocorrelation in Figure 5.3 exhibits the presence of a dominant auto-regressive process (Sudheer *et* al., 2002). Similarly, the partial auto-correlation function in Figure 5.4 shows significant correlation up to lag three, and decays down to small values further, the correlation at lag three is relatively low compared to correlation at lag two. The above analysis of partial- and auto-correlation functions suggests that incorporating flow values up to two days lag in the input vector will be sufficient to model the runoff process. Based on the results of the above analysis, five significant input variables were identified, which form the input vector to the BANN models. The neurons in the input layer therefore consisted of the total rainfall at times t, t-1, and t-2 (P_t, P_{t-1}, and P_{t-2}) and the observed discharges at times t-1 and t-2 (Q_{t-1} and Q_{t-2}).

Two separate BANN models were developed depending upon the manner in which they were trained. The first black-box ANN model (called BANN-BP) employed the popular BP training algorithm using batch learning with momentum factor for its training. The value of learning coefficient of 0.01 and momentum correction factor of 0.075 was employed while training BANN-BP models. Therefore, the architecture of BANN-BP model in the form 5-N-1 was investigated in which N represents the number

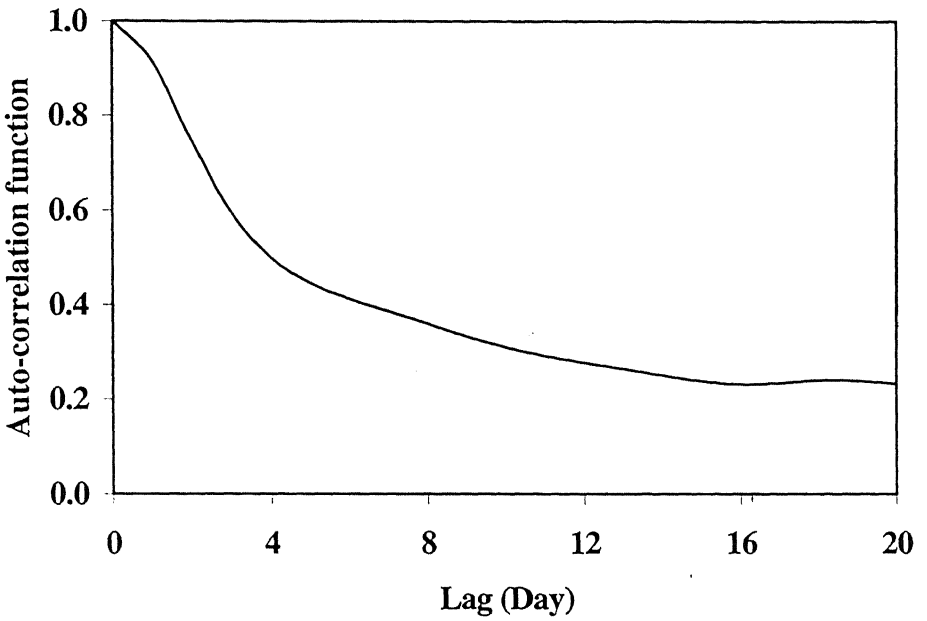


Figure 5.3: Autocorrelation Function of Runoff Series

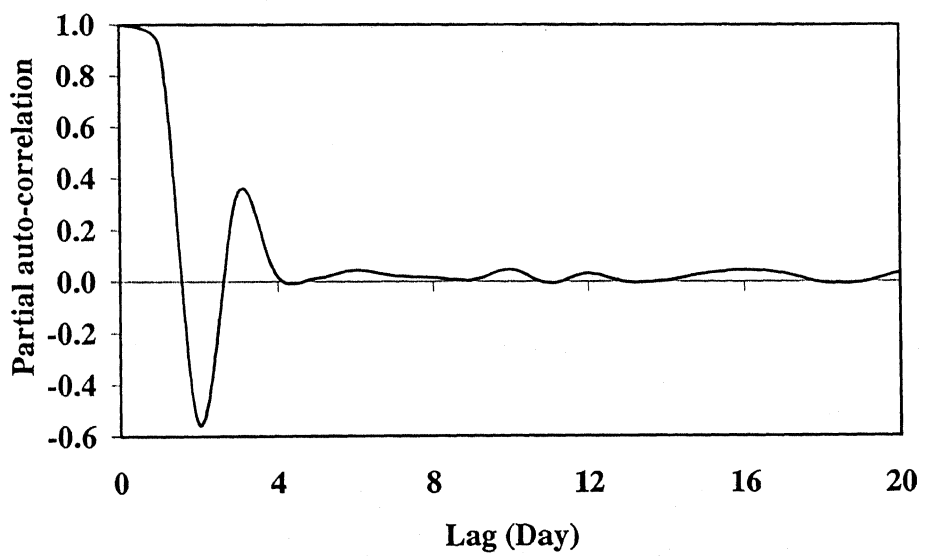


Figure 5.4: Partial Auto-correlation Function of Runoff Series

of hidden neurons. For each set of hidden neurons, the BP algorithm was used to minimize the total error at the output layer. These individual models fits to the calibration data was evaluated using two residual statistics for getting the maximum N-S efficiency (E) and for the minimum AARE to consider the efficiency and effectiveness of the model, respectively. The details about the error statistics are discussed in detail in section 4.5.3. The number of hidden neurons was determined to be four from the trial and error procedure. The second black-box model employed *elitist real-coded* (RGA) for its training

5.2.2 Training of BANN Models using *Elitist R*GA

Essentially, rainfall-runoff modeling using ANNs is equivalent to approximating a multi-dimensional function between inputs and output using the training data set. The popular BP algorithm (Rumelhart et al, 1986) incorporates gradient-descent strategy for calibrating parameters of the ANN model. If the search space consists of two or more dimensions, the gradient-descent strategy may get caught in repeating cycles, where the same local minima solution is found repeatedly (Hassoun 1999). Major limitation of ANN models for streamflow prediction appears to be in prediction over a wide range of stream flows (ASCE task committee, 2000 b). Another aspect, in a continuous rainfall-runoff modeling is that there may be large differential amplitudes of the solutions targeted at each and every output. This causes the error surface to be discontinuous and flat in certain regions of interest. In these situations, change in the shape of the solution surface during training is very slow in the direction where the training domain is broad. Generally, gradient-based techniques are inefficient to face such situations.

Genetic algorithm (GA) is a global search method that does not require the gradient information, and locates globally optimum solution (i.e., global minima) of a given function based on the mechanics of natural selection and natural genetics. One of the apparent distinguishing features of GAs is their effective implementation of parallel multipoint search. GA works by maintaining a balance between exploration and exploitation of search space with guided search process to converge to a global minimum. There can be various ways of using GA based optimization in neural networks. The most evident way is to use GA to search the weight space of an ANN with a predefined architecture. The use of GA based learning methods may be justified for learning tasks that require ANNs with hidden neurons for a non-linear function approximation (Hassoun 1999), which is the case in the problem under consideration. The binary representation traditionally used in GA has some drawbacks (chapter 3) when applied to multi-dimensional high-precision numerical problems. Empirical evidence suggests that different choices/combination of fitness functions and encoding schemes can have significant effect on the GA's convergence time and solution quality (Bäck, 1993).

The floating-point representation of the variables in decimal system results into RGA. The details of *elitist* RGA and its efficiency and effectiveness in finding the global minimum are discussed in detail in Chapter 3 and Chapter 4. The second black-box ANN model developed in this study (called BANN-GA model) employed *elitist* RGA for its training. Further, the structure of the BANN-GA model was kept same as that of the BANN-BP models. The optimum parameter set used to implement *elitist* RGA for training of BANN-GA structure was $P_c = 0.9$, $P_m = 0.01$, $\eta_c = 2$, and $\eta_m = 20$. For consistency and fair comparison, the objective function used in both BANN-BP and

BANN-GA models was same. The flow chart for training of the BANN-GA model using *elitist* RGA is shown in Figure 5.5.

Though the global optimal solution can never be guaranteed, the search for optimal solutions were carried out using ten different runs for BANN-GA model developed in this study in an attempt to obtain global optimal solution. Once the training of both black box models was completed, the calibrated model structures were then used to calculate various performance evaluation indices using both training and testing data sets.

5.2.3 Discussion of Results from BANN Models

The results from the both BANN-BP and BANN-GA models in terms of various performance evaluation indices during both training and testing data sets are presented in Table 5.1. The discussion of results from BANN models presented in two sections. The first section discusses the comparison of BANN-BP and BANN-GA models; whereas, the second section compares BANN models with the CRR Models

5.2.3.1 Results from BANN Models

It is clear from Table 5.1 that the performance of BANN-GA model is better than that of BANN-BP model in terms of all TS, AARE, NMBE, and %MF statistics during both training and testing data sets and in terms of R, E, NRMSE during testing data set; and the performances of both models is comparable in other cases. The minimum AAREs of 22.39% and 23.92% were obtained from the BANN-GA model during training and testing, respectively. Further, about 90% of the training data points estimated from BANN-GA model during training had ARE level of less than 50; but only about 70% of

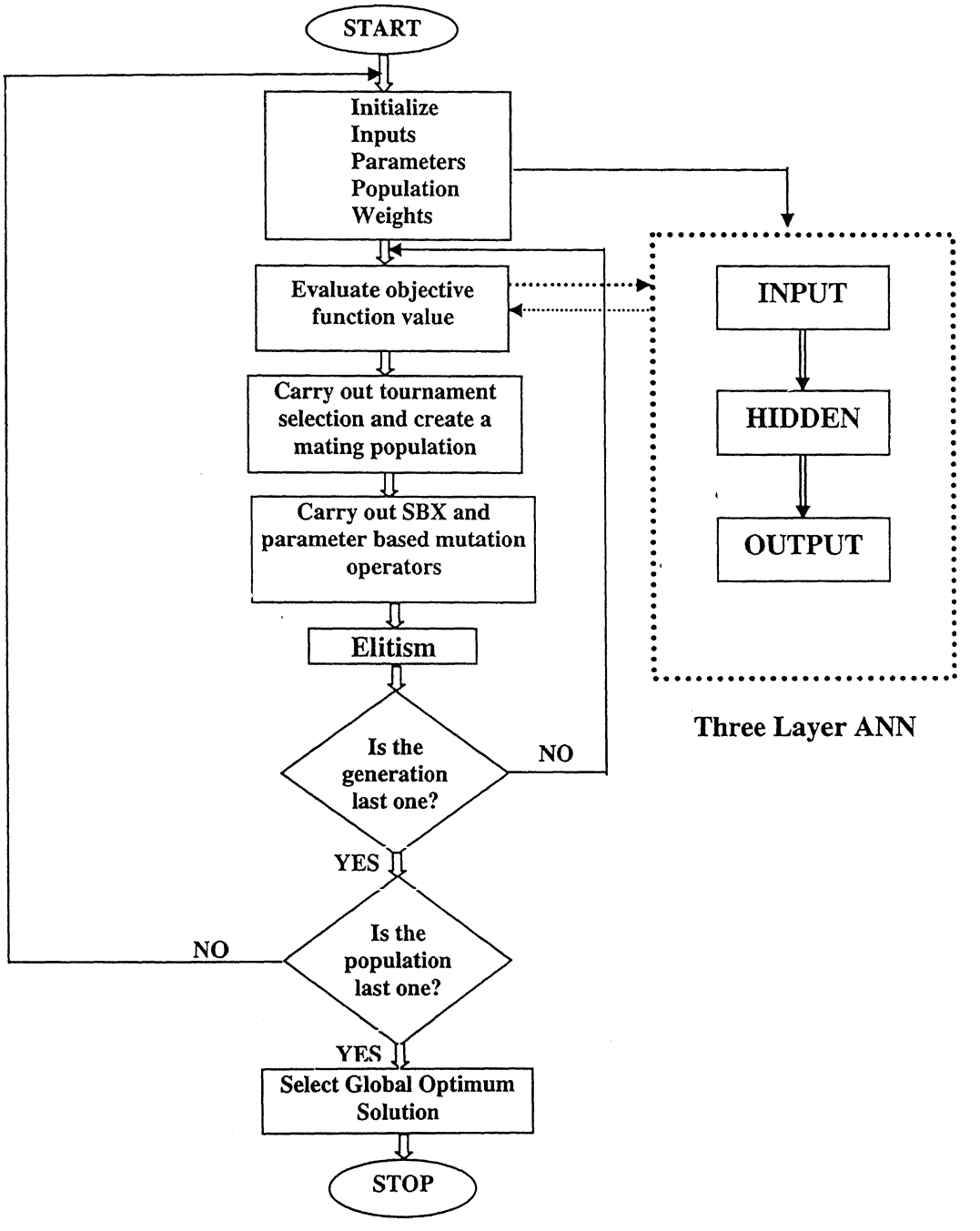


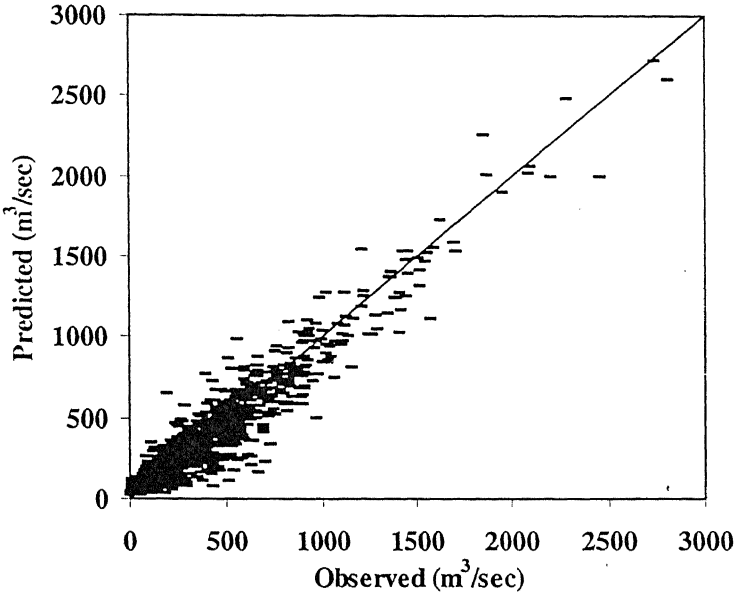
Figure 5.5: Flow Chart for Training ANN using *Elitist Real-Coded* GA

Table 5.1: Performance Evaluation Indices from BANN and GANN Models

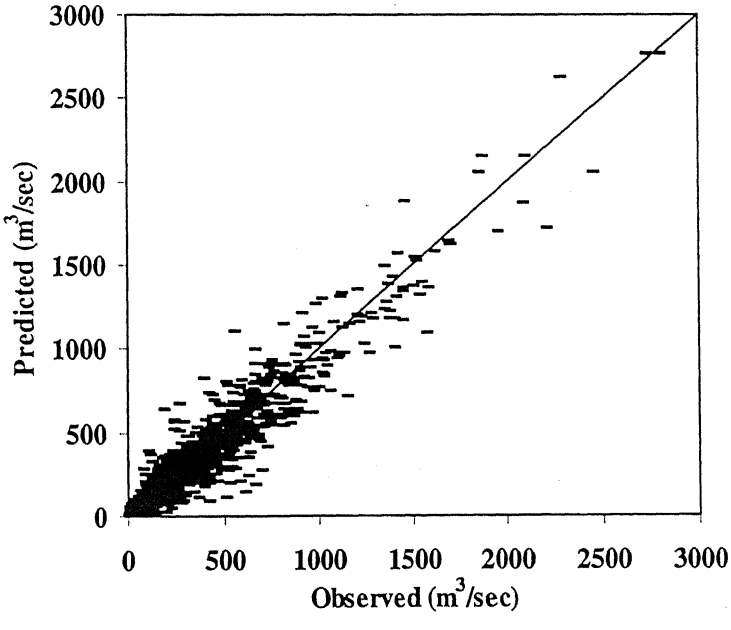
Model	TS1	TS5	TS25	TS50	TS100	AARE	R	E	NMBE(%)	NRMSE	%MF	E_{per}
During Training												
BANN-BP	2.97	13.59	53.47	70.42	83.15	54.45	0.9770	0.9544	1.517	0.347	-6.03	0.731
BANN-GA	4.36	21.10	70.57	90.05	97.91	22.39	0.9767	0.9540	-0.048	0.349	-2.14	0.728
GANN-BP	3.03	14.75	52.94	70.49	83.29	52.20	0.9770	0.9553	1.551	0.344	-4.53	0.736
GANN-GA	5.37	23.06	72.36	91.46	97.99	21.58	0.9773	0.9551	-0.064	0.344	-6.70	0.735
During Testing												
BANN-BP	2.68	12.96	54.34	70.40	80.57	66.78	0.9700	0.9410	0.747	0.388	-7.18	0.687
BANN-GA	4.00	20.33	67.84	87.65	97.85	23.92	0.9701	0.9412	-0.172	0.387	-1.62	0.689
GANN-BP	3.27	14.98	55.63	70.61	81.04	62.48	0.9700	0.9413	-0.300	0.387	-2.98	0.689
GANN-GA	4.64	22.50	69.72	89.02	98.18	23.09	0.9704	0.9416	-0.963	0.385	-4.82	0.691

the training data points estimated from BANN-BP model had ARE level of less than 50% (see TS50 in Table 5.1). The superiority of the BANN-GA model over the BANN-BP model can be assessed by observing TS for other ARE levels also during both training and testing data sets. Overall, it can be said that even though the performance of the two models is similar with respect to efficiency in modeling in terms of R, E, and NRMSE statistics; the BANN-GA model performs significantly better with respect to effectiveness in predicting flows accurately in terms of AARE, TS, NMBE, E_{per} statistics.

The results in graphical form from BANN models are presented in Figure 5.6 through Figure 5.9. Figure 5.6 show the observed and predicted flow values (m^3/sec) against each other as scatter plots for the testing data set of 13-years (1977-1989) from the two black box models; whereas, Figure 5.7 and Figure 5.8 show the observed and predicted flow values against time (in days) for the dry and wet years 1986 and 1989 from BANN-BP and BANN-GA models, respectively. It can be noticed from Figure 5.6 that the observed flow values match very well with the predicted flow values obtained from the two models. The observed and predicted values from the BANN-GA model appear to be more close to 45° line as compared to those from the BANN-BP model. Further, a close examination of Figure 5.7 reveals that the ANN models trained using BP training algorithm tend to over-estimate lower magnitude flows. This is magnified in the Figure 5.9, where part of the estimated dry year data both from BANN-BP and BANN-GA models is shown. The problem of over-estimating the lower magnitude flows by in BANN-BP model has been clearly overcome by the BANN-GA model, which can be verified through Figure 5.9 for the year 1986, where the lower magnitude flows are estimated very well from BANN-GA model. However, it was can be observed from the

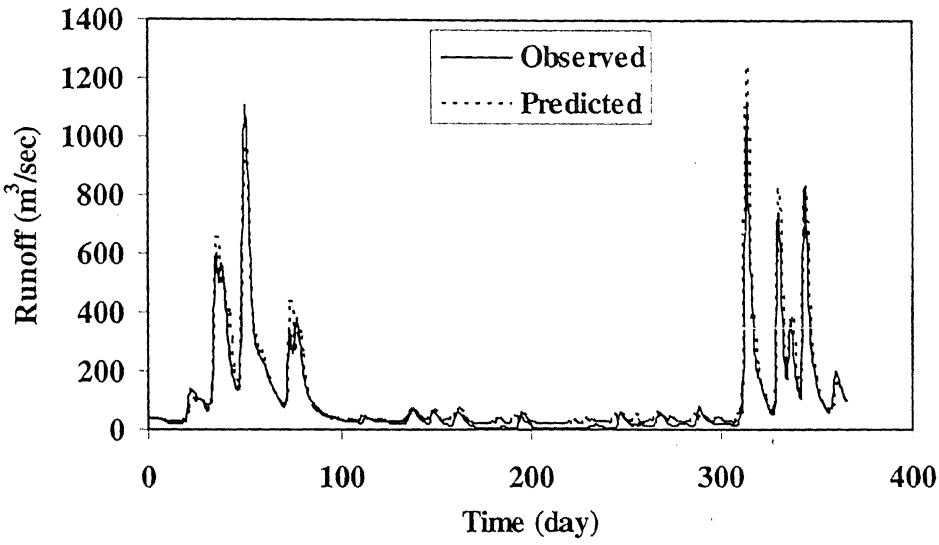


(a) From BANN-BP Model

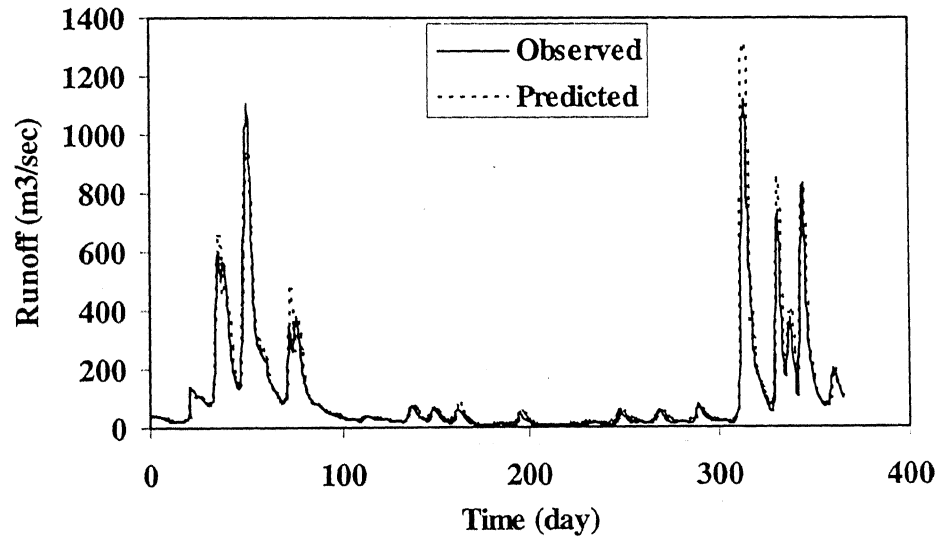


(b) From BANN-GA Model

Figure 5.6: Observed and Predicted Flows during Testing

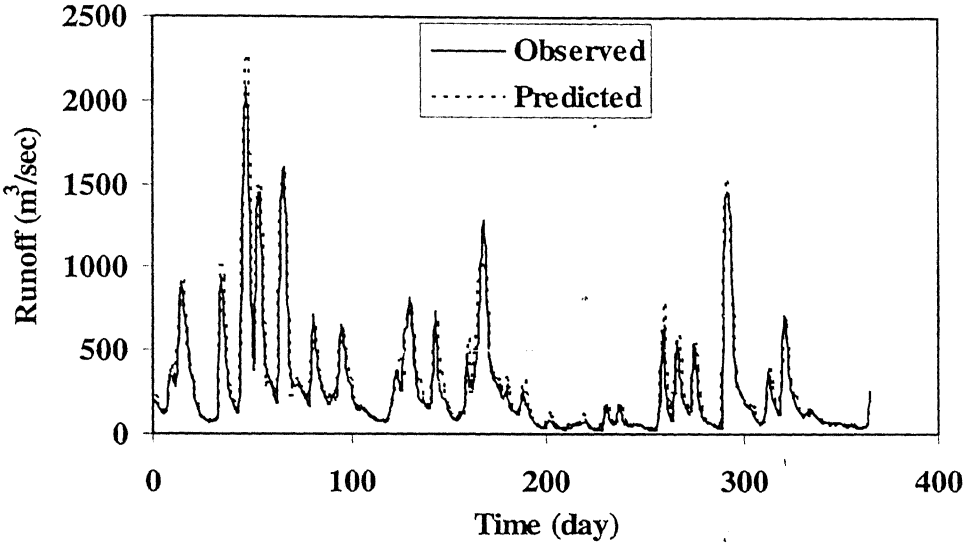


(a) From BANN-BP Model

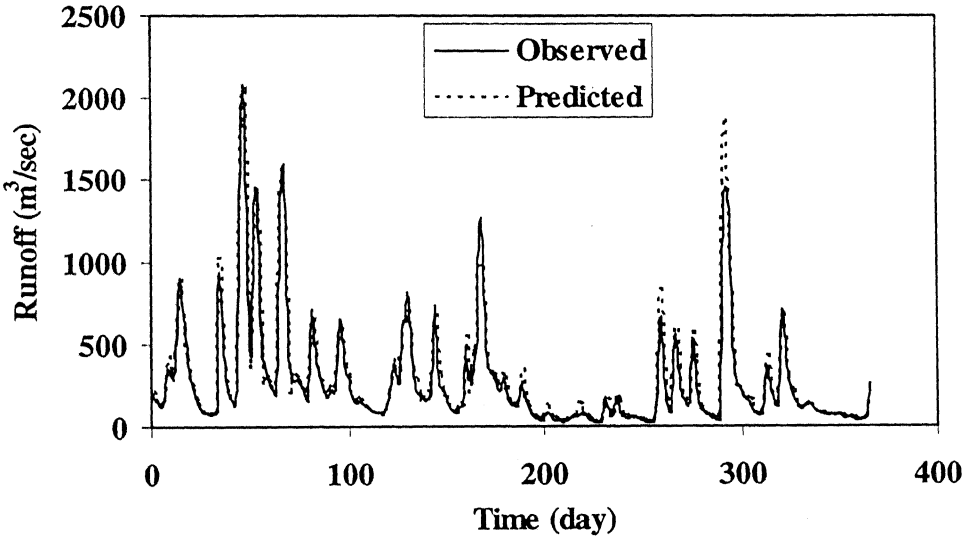


(b) From BANN-GA Model

Figure 5.7: Observed and Predicted Flow in the Dry Year 1986

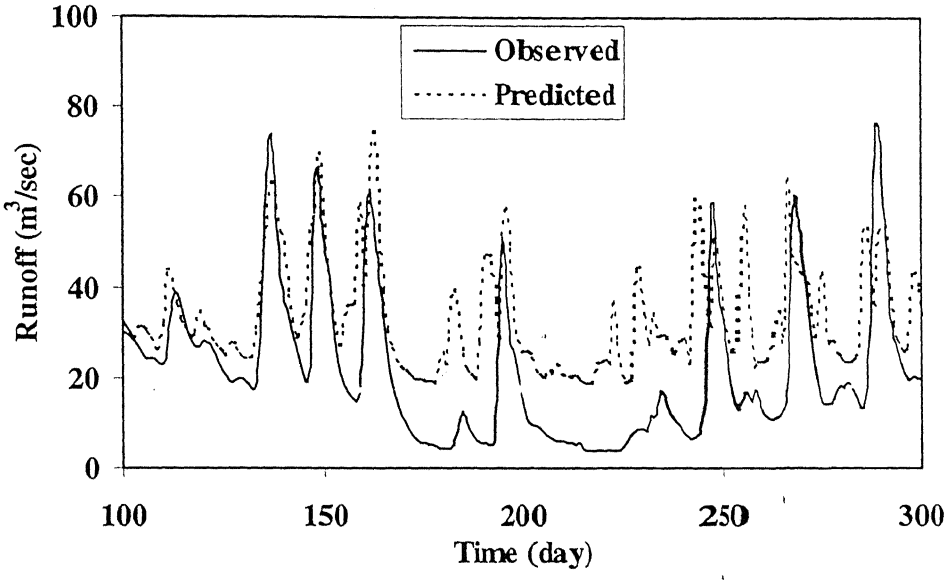


(a) From BANN-BP Model

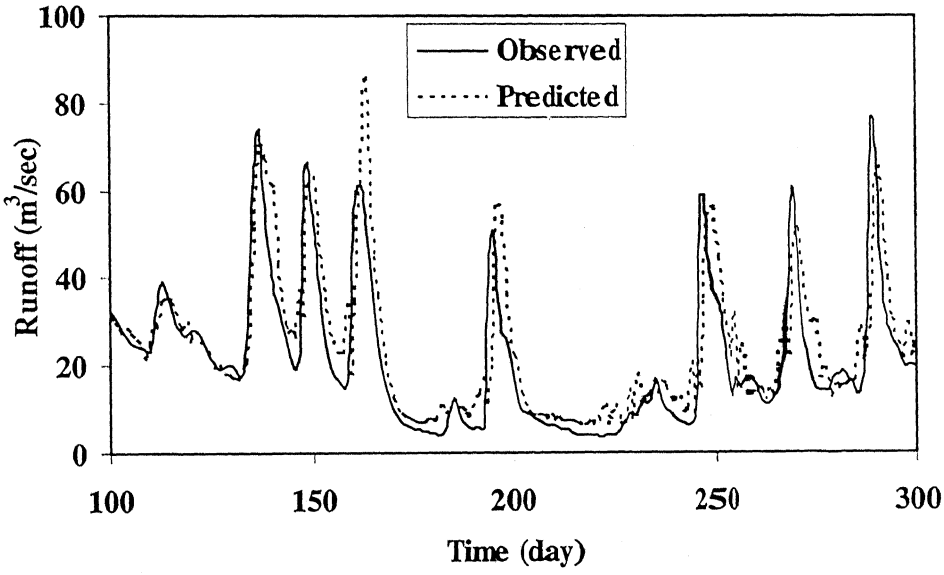


(b) From BANN-GA Model

Figure 5.8: Observed and Predicted Flow in the Wet Year 1989



(a) From BANN-BP Model



(b) From BANN-GA Model

Figure 5.9: Observed and Predicted Low Flows in 1986

Figure 5.8 for the wet year 1989 that while the BANN-GA model overestimated one of the peaks compared to the BANN-BP model close to 300th day, BANN-BP model overestimated the peak close to 50th day.

The discussions of results presented above, however, are based on graphical results from only two years of testing data set (1986 and 1989), and may not give true representation of the performance of models considering total data set. In order to assess the relative strengths and weaknesses of different models in accurately predicting different magnitude flows, a more detailed and close examination of the results spanning over the whole range of training and testing data sets is needed. This can be accomplished by carrying out a 'partitioning analysis' on the results from both training and testing data sets to examine the efficiency and effectiveness of various models in predicting low, medium, and high magnitude flows accurately by computing the AARE and threshold statistics for low, medium, and high magnitude flows separately. This 'partitioning analysis' of the results was carried out by subdividing the total data results from both training and testing data sets into three categories. The partitioning of the data was based on the relative spread of the flows from the mean, and a careful examination of the distribution of flows in terms of different statistical measures. The statistical properties of the data are: mean (μ) = 151.85 m^3/s, standard deviation (σ) = 239.9 m^3/s, and coefficient of variation (C_v) = 1.58. It is clear that the flow data under consideration have very high degree of variability, and only a robust model structure will be able to capture such a high variability in the data set. The flows having magnitudes less than 151.85 m^3/s ($x < \mu$) were classified as low flows, flows having magnitudes between 151.85 m^3/s and 631.65 m^3/s ($\mu \leq x \leq \mu + 2\sigma$) were classified as medium flows, and the flows having magnitudes greater than 631.65 m^3/s ($x > \mu + 2\sigma$)

were classified as high flows. The number of data points in low, medium, and high magnitude flow categories for training and testing data sets are presented in Table 5.2. It is clear from Table 5.2 that the majority of data points are in low magnitude flow category both during training and testing data sets (approximately 72%). Therefore, the overall performance of any model in estimating flows will be dictated by its effectiveness in accurately estimating the low magnitude flows.

The results of the partitioning analysis for BANN models are presented in Table 5.3 and Table 5.4 for training and testing data sets, respectively. The tables present the TS, AARE, R, E, NMBE, and NRMSE statistics for low, medium, and high magnitude flows separately. A significant improvement in modeling the low flows can be observed from Table 5.3 and Table 5.4 in terms of all the statistics. For example, the AARE for low magnitude flows from BANN-BP model of 69.49% (very poor) improves to 24.82% (very good) from BANN-GA model during training. The TS100 of 76.47% from BANN-BP model improves to 97.22% from BANN-GA model during training; meaning more than 97% of the predicted data points during training had ARE level of less than 100%.

Similar improvements in estimating low flows can be observed during testing data set. Also the improvement in threshold statistics and AARE are marginal for medium magnitude flows and are not apparent for high magnitude flows. Similarly, significant improvements in R, E, NMBE, and NRMSE statistics can also be observed for low flows, during both training and testing data sets. It must be noted that the performance in estimating the high and medium magnitude flows is reasonably good from both the models. The modeling of low flows is actually problematic using BP algorithm also

Table 5.2: Number of Data Points in Low, Medium, and High Magnitude Flow Categories

Category	Training Data		Testing Data	
	No. of Points	Percentage	No. of Points	Percentage
Low	3387	71.35	3446	72.61
Medium	1020	21.49	961	20.25
High	340	07.16	339	07.14
Total	4747	100.0	4746	100.0

Table 5.3: Performance Statistics for Low Medium & High Magnitude Flows from BANN and GANN Models during Training Data

Model	TS1	TS5	TS25	TS50	TS100	AARE	R	E	NMBE	NRMSE
Low Magnitude Flows										
BANN-BP	2.27	9.89	43.54	60.35	76.47	69.49	0.8629	0.6765	16.45	0.506
BANN-GA	4.04	19.25	66.90	88.04	97.22	24.82	0.8961	0.7490	6.81	0.446
GANN-BP	1.89	10.51	43.05	60.67	76.68	66.44	0.8794	0.7103	16.85	0.479
GANN-GA	4.81	20.93	69.76	89.99	97.34	23.72	0.9022	0.7568	7.78	0.439
CRR Model-II	8.38	34.35	81.44	92.53	96.99	20.90	0.8599	0.6434	4.10	0.532
Medium Magnitude Flows										
BANN-BP	4.41	22.06	76.27	94.90	99.71	17.88	0.8476	0.6668	16.42	1.092
BANN-GA	5.88	26.47	78.72	94.41	99.50	16.79	0.8823	0.7431	6.11	0.959
GANN-BP	6.18	24.12	75.59	94.12	99.71	17.58	0.8669	0.7017	16.95	1.033
GANN-GA	6.86	29.21	76.56	94.21	99.50	16.92	0.8934	0.7522	7.38	0.942
CRR Model-II	8.15	33.76	80.27	92.05	97.55	19.45	0.8295	0.6205	3.20	1.167
High Magnitude Flows										
BANN-BP	5.59	25.00	83.82	97.35	100.00	14.22	0.8136	0.6593	7.61	1.589
BANN-GA	2.94	23.52	82.64	97.05	100.00	15.00	0.8341	0.7326	4.35	1.408
GANN-BP	5.00	28.82	83.53	97.35	100.00	14.10	0.8395	0.6955	9.68	1.502
GANN-GA	6.47	25.88	85.58	97.94	100.00	14.15	0.8549	0.7437	6.59	1.378
CRR Model-II	10.29	38.53	82.06	92.06	97.65	19.80	0.7483	0.5920	3.60	1.721

Table 5.4: Performance Statistics for Low Medium & High Magnitude Flows from BANN and GANN Models during Testing

Model	TS1	TS5	TS25	TS50	TS100	AARE	R	E	NMBE	NRMSE
Low Magnitude Flows										
BANN-BP	2.06	10.10	46.69	61.52	73.30	85.00	0.8734	0.7312	13.14	0.438
BANN-GA	3.80	19.47	64.39	85.08	97.21	26.31	0.9085	0.7935	5.78	0.384
GANN-BP	2.52	12.48	48.20	61.84	74.03	79.10	0.8885	0.7632	12.67	0.417
GANN-GA	4.52	21.35	67.00	86.99	97.73	25.26	0.9087	0.7847	6.08	0.392
CRR Model-II	10.70	39.62	82.13	91.60	96.68	22.62	0.8768	0.7087	3.70	0.462
Medium Magnitude Flows										
BANN-BP	3.95	20.19	73.15	93.44	99.79	19.08	0.8452	0.7168	10.54	0.877
BANN-GA	4.78	23.62	76.79	93.65	99.37	17.77	0.8785	0.7746	5.38	0.782
GANN-BP	5.31	21.64	74.82	93.24	99.48	18.81	0.8451	0.7324	13.26	0.837
GANN-GA	5.51	28.51	76.27	93.54	99.16	17.53	0.8779	0.7599	7.89	0.807
CRR Model-II	8.12	35.28	79.60	90.63	96.05	25.26	0.8194	0.6647	4.90	0.939
High Magnitude Flows										
BANN-BP	5.30	21.53	78.76	95.28	100.00	16.76	0.8362	0.7128	3.31	1.266
BANN-GA	3.83	19.76	77.58	96.75	100.00	16.95	0.8725	0.7699	2.84	1.133
GANN-BP	5.01	21.53	76.70	95.57	100.00	17.36	0.8236	0.7203	7.58	1.222
GANN-GA	3.83	17.10	78.76	96.75	100.00	16.85	0.8705	0.7535	5.80	1.172
CRR Model-II	7.67	32.74	80.53	90.86	96.17	22.60	0.8321	0.6558	4.60	1.358

pointed out by other researchers in the past (Hsu et al, 1995; Sajikumar and Thandaveswara, 1999; Tokar and Markus, 2000), which can be overcome by employing other training methods to train the ANN rainfall-runoff models. Therefore, based on the results obtained in terms of BANN models, it can be said that overall prediction capability can be improved by employing *elitist* RGA instead of popular BP method in training the ANN rainfall-runoff models.

5.2.3.2 Comparison of BANN and CRR Models

Comparing the results of BANN models from Table 5.1 and those from the two CRR models from Table 4.4, it can be observed that the performance of the BANN models is superior to the CRR models in efficiency in modeling the complex, dynamic, and non-linear rainfall-runoff process in terms of the R, E, NMBE, NRMSE and E_{per} statistics. For example, considering the error statistics during calibration, the value of E_{per} statistic from the BANN-BP and BANN-GA models, 0.731 and 0.728, respectively, is much higher than those from the two CRR models of 0.109 and 0.319 meaning the BANN models are much better than the simple persistence model in comparison to the two CRR models developed in this study. Further, the ability of the two BANN models in explaining the variance in the rainfall and runoff data in terms of the coefficient of efficiency (E) during calibration is in excess of 0.95; whereas, the value of E from the CRR models is in the vicinity of 0.85 only. Similar trends of the superiority of BANN models in efficiency in modeling over the CRR models can be noticed during the validation data set. However, it is interesting to note that the performance of the two CRR models is better than the BANN models measured by the effectiveness in predicting the flows in terms of the TS and AARE statistics. It is especially interesting to note that the performance of the BANN model trained using BP training method

(BANN-BP model) is very poor in terms of TS and AARE statistics compared to the two CRR models both during calibration and validation data sets. For example, the values of AARE from the CRR Model-II of 21.29% and 22.95% is much better than those from the BANN-BP model of 54.45% and 66.78% during calibration and validation data sets, respectively. The poor performance of the BANN-BP model compared to the two CRR models in terms of effectiveness in prediction can be attributed to the inability of the BP algorithm to model the low magnitude flows properly, also pointed out in literature in the past by other researchers (Hsu et al. 1995; Tokar and Markus 2000). In order to verify this, a partitioning analysis the results was carried out by dividing the error statistics corresponding to the low, medium, and high magnitude flows into different categories. The results of the partitioning analysis are provided in Table 5.3 and Table 5.4 during calibration and validation data sets, respectively. It can be noted that the CRR Model-II is able to predict the low magnitude flows with much better effectiveness in terms of TS and AARE statistics as compared to the BANN-BP model both during calibration and validation data sets. Another possible reason of the CRR models performing better than the BANN models in predicting flows may be due to the fact that the CRR models employ a simple concept of flow recession to model the falling limb of the flow hydrograph; whereas, the BANN models attempt to capture different dynamics of the hydrological processes inherent in the rising and falling limbs of the flow hydrograph using a single feed-forward ANN. This indicates the need to decompose the input output data space into different classes corresponding to different segments of a flow hydrograph having different dynamics of the rainfall-runoff relationships. Further, the performance of the BANN model trained using the *elitist* RGA (BANN-GA model) is comparable to the better of the two CRR models (i.e.

validation data sets. This highlights the superiority of the *elitist* RGA over the popular BP algorithm in achieving a better generalization of the complex, dynamic, and non-linear rainfall-runoff relationship while developing the ANN rainfall-runoff models.

Further, a close examination of the scatter plots of the observed and predicted flows for the validation data set of 13-years from the two CRR models (Figure 4.11) and the two BANN models (Figure 5.6) indicates an interesting observation about the nature of the errors in prediction of flows from the two types of models. It can be noticed from Figure 4.11 that the scatter points corresponding to the observed and predicted flows from the CRR models deviate from the 45-degree line in a systematic manner. In other words, the scatter points tend to lie above the 45-degree line indicating consistent over-prediction. This trend is more pronounced from CRR Model-I as compared to the CRR Model-II. The consistent over-prediction, which can also be verified by the high magnitudes of NMBE% statistics of 8.951% and 9.462% during calibration and validation from the CRR Model-I, respectively, indicates the nature of the errors from the CRR models to be systematic, which is caused either by systematic errors in the data set or the model errors. Further, the close examination of Figure 5.6 indicates that the scatter points are evenly distributed around the 45-degree line indicating that the residual errors from the BANN models are random in nature. This also excludes the possibility of a systematic error being present in the rainfall-runoff data, indicating the presence of model structure errors in the CRR Model-I, which employed a linear reservoir to model the surface flow component. The more or less random nature of the residual errors from the CRR Model-II justifies the use of two conceptual elements, namely, linear channel element and the non-reservoir element, in the surface flow component to model the complex, dynamic, and non-linear rainfall-runoff process.

Finally, it is clear that the performance of the BANN-GA model is better than the CRR Model-II in efficiency in modeling in terms of R, E, NMBE, NRMSE, and E_{per} statistics and is similar in terms of effectiveness in prediction in terms of TS and AARE statistics during both calibration and validation data sets. Therefore, it can be said that the black box ANN model trained using the *elitist* RGA (BANN-GA) is the best among the four models discussed so far in this study when evaluating the performances of all the models in terms of the eight performance evaluation indices considered in this study for the purpose of modeling the complex, dynamic, and non-linear rainfall-runoff process.

5.3 *Grey-Box* ANN (GANN) Models

The transformation of a sequence of total rainfalls into a series of discharge hydrographs is an extremely complex, dynamic, and non-linear phenomenon, which involves various components of the hydrologic cycle and physical variables. In the past, attempts on modeling different components of the hydrologic system have focused either on a white (or transparent) box type model through which the details of the physical process are visible (e.g. deterministic or conceptual models) or a black-box type model through which the details of the physical process are invisible. The concept of transparent and black-box models is shown in Figure 5.10(a) and Figure 5.10(b), respectively. The modeling of rainfall-runoff process using ANNs is considered as black-box models of non-linear systems theoretic type. The ANN rainfall-runoff models have achieved some degree of success in stream flow prediction as they produce for quick and reliable forecasts (ASCE task committee 2000, b). A distinct advantage of an ANN is that it learns unknown relationship through a process of training between inputs and outputs. In contrast, the advantage of conceptual approach is that it considers the inherent physical processes in a systematic manner while developing rainfall-runoff

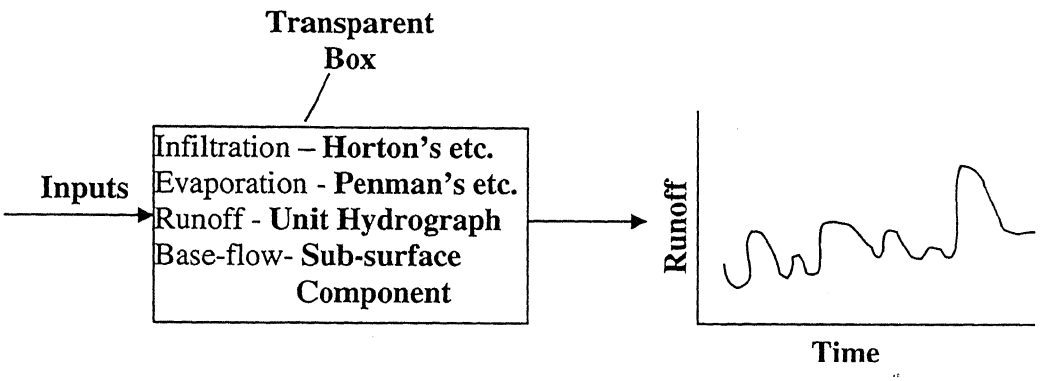


Figure 5.10(a): Transparent Box Model

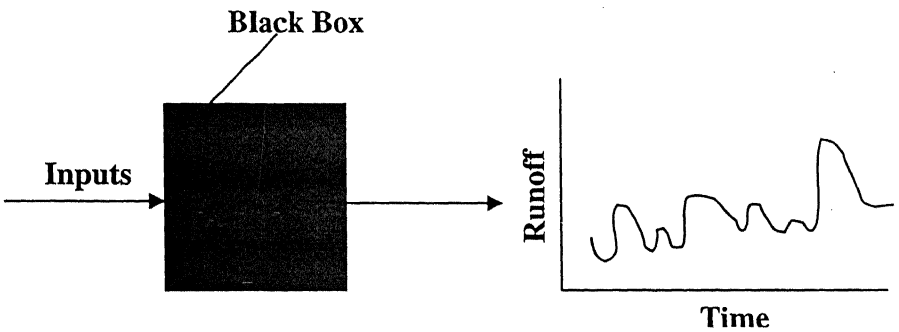


Figure 5.10 (b): Black-Box Model

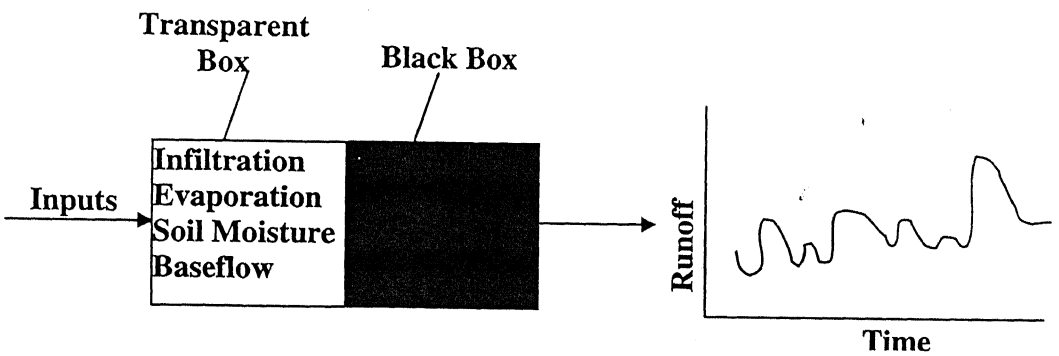


Figure 5.10 (c): Grey-Box Model

models. Ideally, it should be possible to develop a new class of models capable of exploiting the advantages of both conceptual and ANN techniques.

In this section, an attempt is made to develop a different type of methodology to model rainfall-runoff process using non-linear system theoretic technique (ANN) and conceptual techniques. Such models are partially transparent, and can be termed as *grey-box* models. This can be achieved through the integration of conceptual and non-linear systems theoretic techniques. The concept of *grey-box* models is depicted in Figure 5.10(c). The *grey-box* models presented in this section are based on the concept that the synthesis of conceptual and non-linear systems theoretic techniques can result in an integrated model that is more effective and efficient in capturing the complex, dynamic, and highly non-linear rainfall-runoff process compared to a model that uses either of these techniques in isolation. Two types of *grey-box* models have been developed in this study. The first type of *grey-box* models integrate the conceptual techniques with the ANN technique and is referred to as the GANN-BP model, while the second type of *grey box* models integrate the deterministic, ANN, and elitist RGA techniques, and is referred to as the GANN-GA model

The *grey-box* model developed in this study uses a conceptual method of computing infiltration and hence effective rainfalls on a continuous basis using Green-Ampt equations, Haan's (1972) method of computing daily expected evapotranspiration, the law of conservation of mass to continuously update the sub-surface storage component, and an ANN technique to model the transformation of effective rainfalls into a series of discharge hydrograph. The parameters of the Green-Ampt equations were determined using the CRR model-II using rainfall and streamflow data taken from the Kentucky

River watershed using the *elitist* RGA as discussed in the Chapter 4. The calibrated values of the Green-Ampt parameters were determined to be K = 0.1999 mm/hour, ψ = 201.021 mm, η = 0.11229, and S = 305.66 mm. Once the parameters were determined, actual incremental infiltration at each time step (ΔFA_t) can be estimated by comparing the potential incremental infiltration computed using the Green-Ampt infiltration equations at each time step with the rainfall values at the corresponding time step. The effective rainfall at each time step can then be calculated by simply subtracting the estimated actual incremental infiltration from the total rainfall at that time step. The basic structure of both the *grey-box* models is same and consists of the following main components: (a) base flow model component (b) infiltration model component, (c) soil moisture accounting (SMA) model component, and (d) the ANN model component. The structure of integrated ANN rainfall-runoff model is shown in Figure 5.11.

The development of the ANN model component of the *grey-box* models was similar to the development of the black box ANN model. The cross-correlation plot between effective rainfall lags and runoff at time 't' (Figure 5.12), and the and auto-correlation function plot of effective rainfall (Figure 5.13) suggested incorporating effective rainfall values up to a lag of 2 to be considered in the input vector for the development of *grey-box* models. The structure of the ANN model component of the *grey-box* model was also of the form 5-N-1. The input vector was similar to that in the BANN model except that the effective rainfall computed using the conceptual techniques replaced the total rainfall.

Two separate *grey-box* models were developed that differed in the method of training the ANNs. The first *grey-box* model employed back-propagation training algorithm

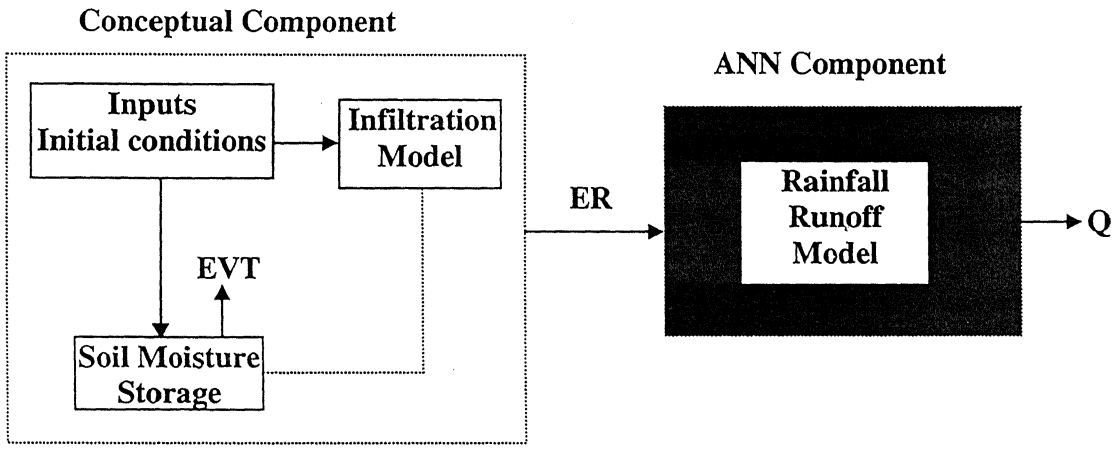


Figure 5.11: Structure of the Grey-Box ANN Rainfall-Runoff Model

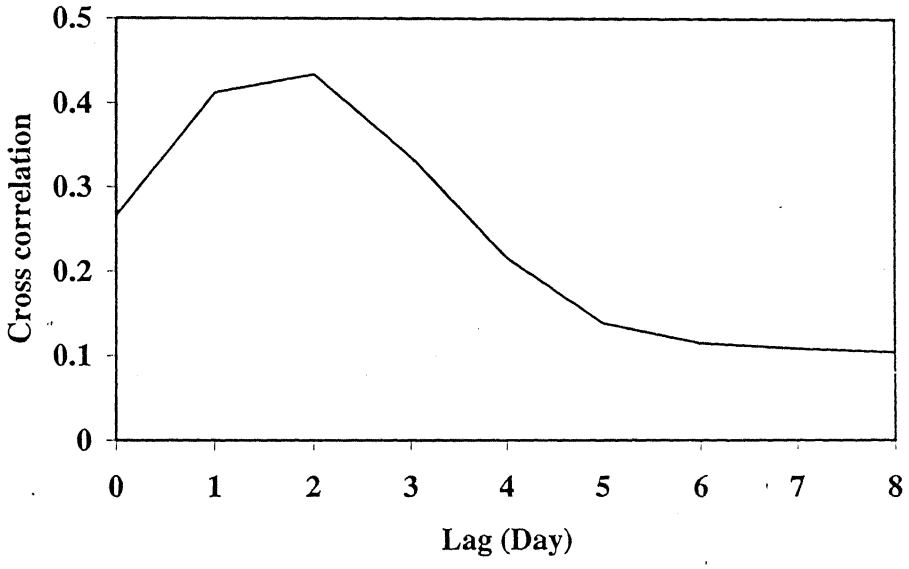


Figure 5.12: Cross Correlation between flow at Time t and Lagged Effective Rainfalls

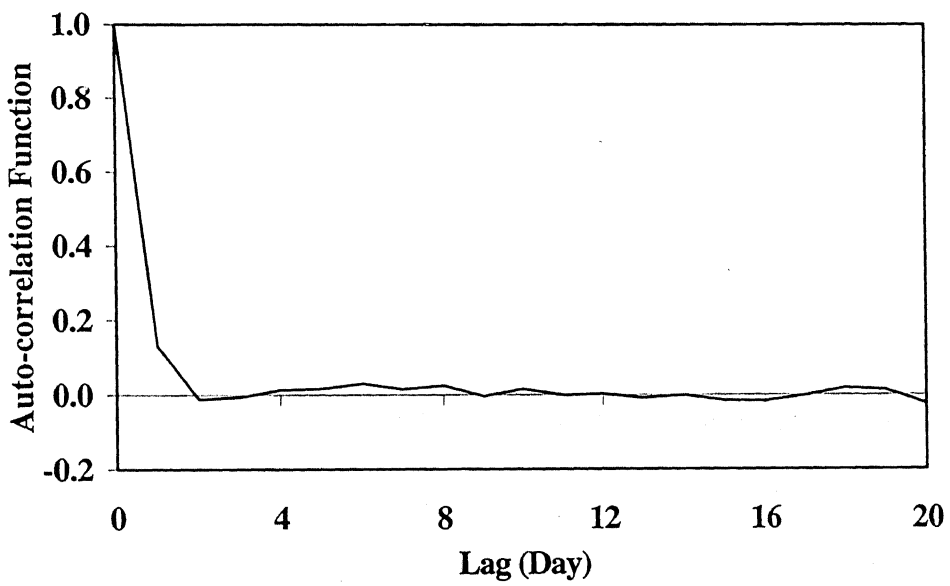


Figure 5.13: Autocorrelation Function of Effective Rainfall Series

(termed GANN-BP model) and the second *grey-box* model employed *elitist* RGA as the training algorithm (termed GANN-GA model). Once the training of both *grey-box* models was complete, they were used to produce streamflow forecasts for the training and testing data sets. The performance of the GANN-BP and GANN-GA models in predicting flow was then evaluated using the eight performance evaluation indices considered in this study.

5.3.1 Discussion of Results from GANN Models

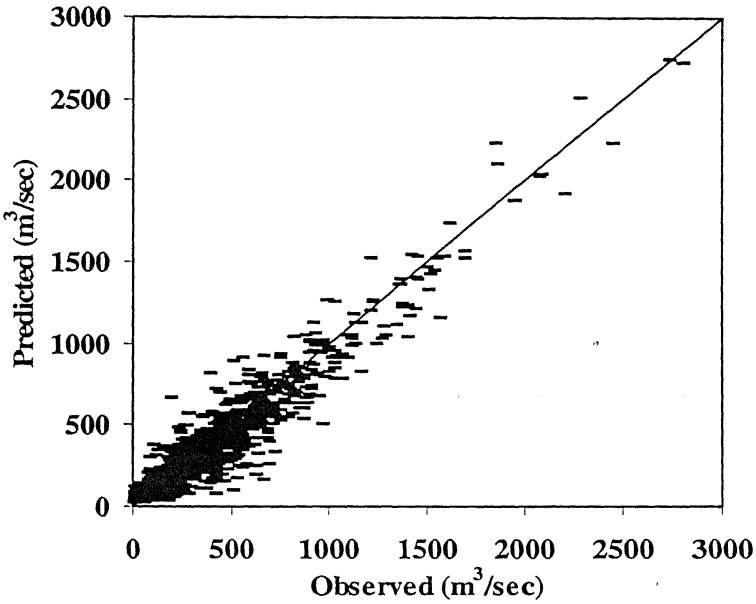
The results obtained from the two *grey-box* ANN models in terms of performance evaluation indices for training and testing data sets considered in this study are presented in Table 5.1. The 'partitioning analysis' of different ranges of flow and their distribution of errors both for training and testing data sets are presented in Table 5.3 and Table 5.4 respectively. The discussion of results from GANN models has been divided into two sections. The first section evaluates the performance of GANN models; whereas the second section compares the performance of BANN and GANN models.

5.3.1.1 Results from GANN Models

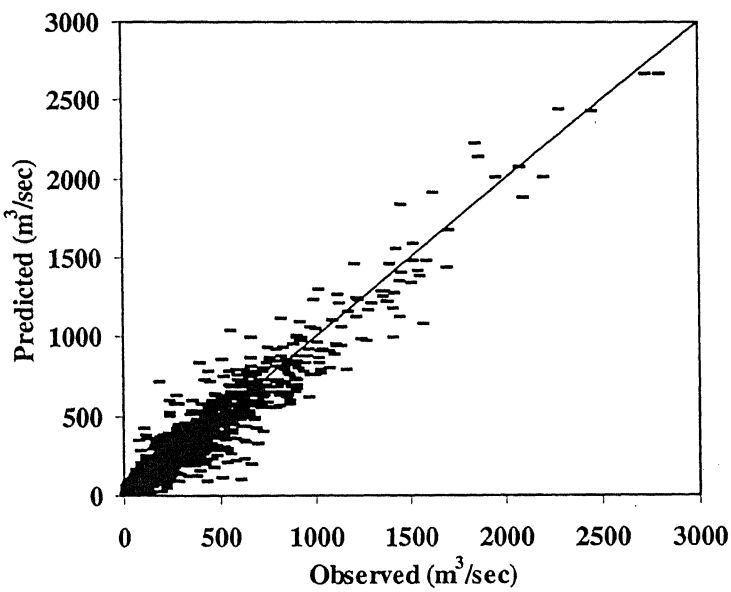
It can be observed from the Table 5.1 that the performance of GANN-GA model is better than the GANN-BP model in terms of all the performance evaluation indices during both training and testing data sets, except for E, %MF, and E_{per} during training, and for NMBE% and %MF during testing, when the performance of GANN-BP model is only slightly better than the GANN-GA model. For example, the value of AARE and R during training from the GANN-GA model of 23.09% and 0.9704 were obtained as against the AARE and R values of 62.48 and 0.9700 from the GANN-BP model.

Looking at the threshold statistics, it can be noted that only 1.82% of total flow values predicted during testing (approximately 4747 data points) from the GANN-GA model had ARE less than 100% as compared to the GANN-BP model, from which 18.96% of the total flow values predicted during testing had ARE less than 100% (see TS100 in Table 5.1). A similar trend can be observed in terms of TS values for other ARE levels during both training and testing data sets. The edge in the performance of the GANN-GA model over number of statistical indices both for the training and verification data sets suggests that the global search strategy used to train ANN for the rainfall-runoff modeling has definite advantage over BP trained models.

The results in graphical form are presented in Figure 5.14 through Figure 5.17. Figure 5.14 shows the observed and predicted flow values (m^3/sec) from the two *grey-box* models as scatter plots for the testing data set. The even distribution of scatter points around 45° line shows that there is no indication of systematic error in the estimation of flow from the models and the residual errors are random in nature. The predicted flows for the dry and wet years of 1986 and 1989 from the two GANN models are presented in Figure 5.15, and Figure 5.16 respectively. The part of the dry year data during summer, as hydrographs, is presented in Figure 5.17 from the GANN-BP and GANN-GA models. It is clear that the predicted flows from GANN-GA model match very well with the observed flows during both dry and wet years. Further, a close visual inspection of the Figure 5.15 shows that the GANN-BP model over estimates the low flows, this magnified in the Figure 5.17. Figure 5.17 shows that the GANN-GA model has superior low flows predicting ability over BP trained integrated ANN model. It can be observed from Figure 5.16 that both the models perform equally well in predicting high magnitude flows. Further, the 'partitioning analysis' also reveals that better error

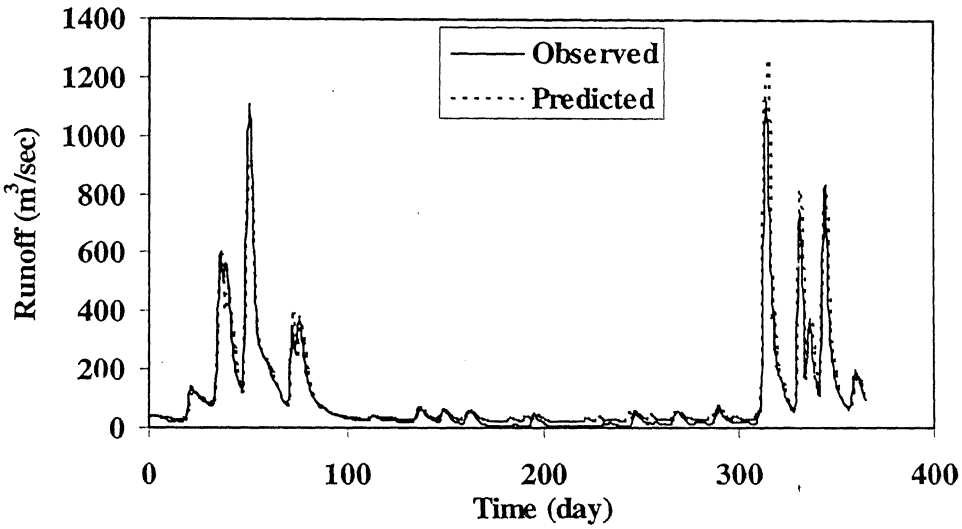


(a) From GANN-BP Model

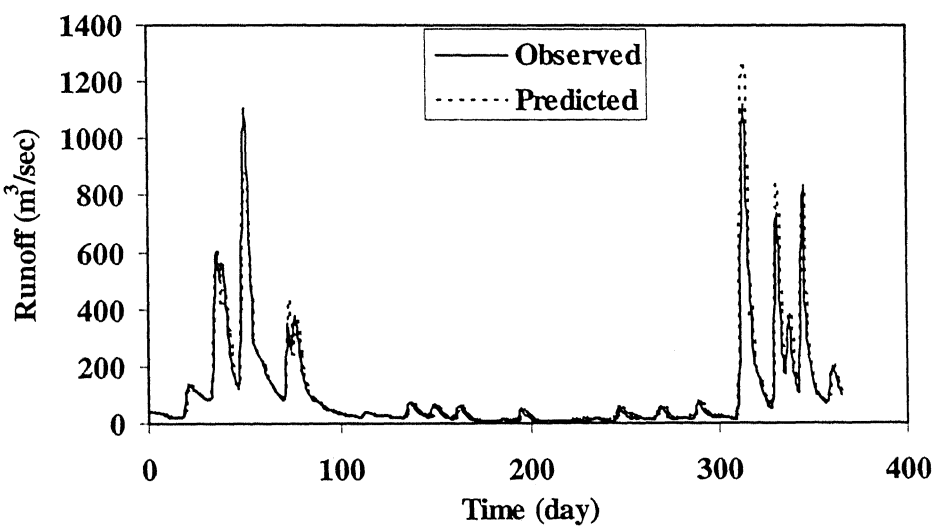


(b) From GANN-GA Model

Figure 5.14: Observed and Predicted Flows during Testing

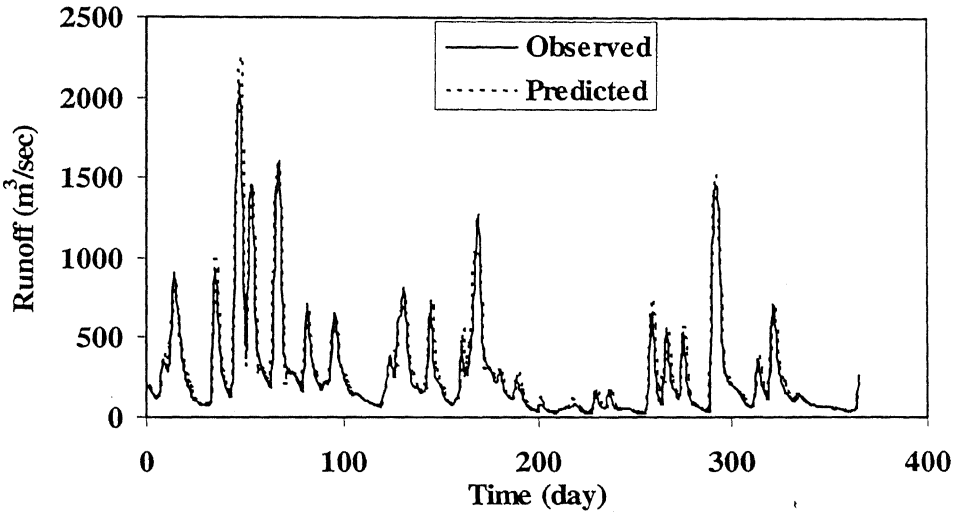


(a) From GANN-BP Model

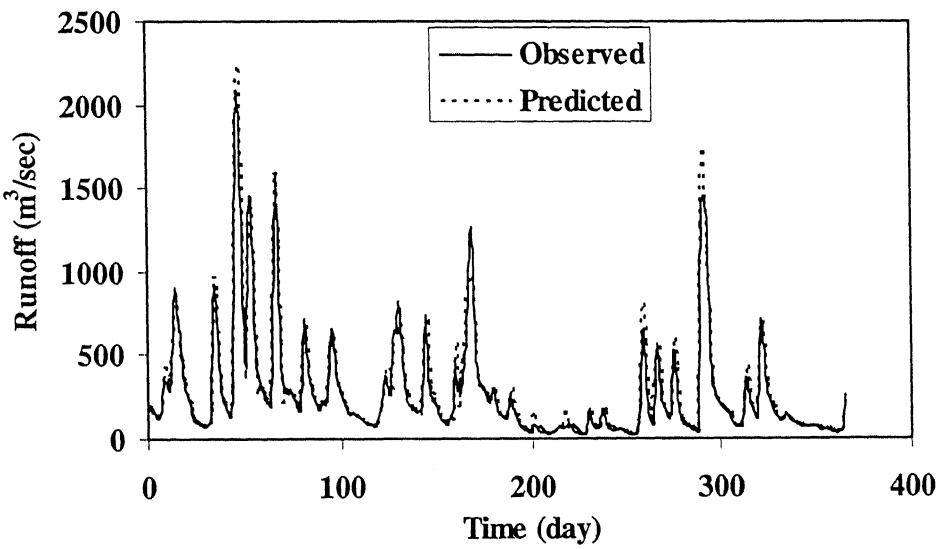


(b) From GANN-GA Model

Figure 5.15: Observed and Predicted Flow in the Dry Year 1986

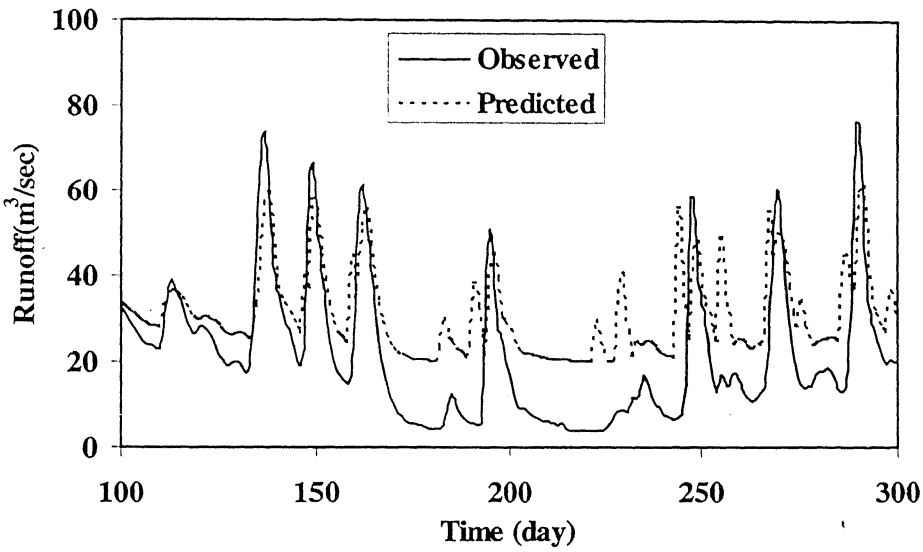


(a) From GANN-BP Model

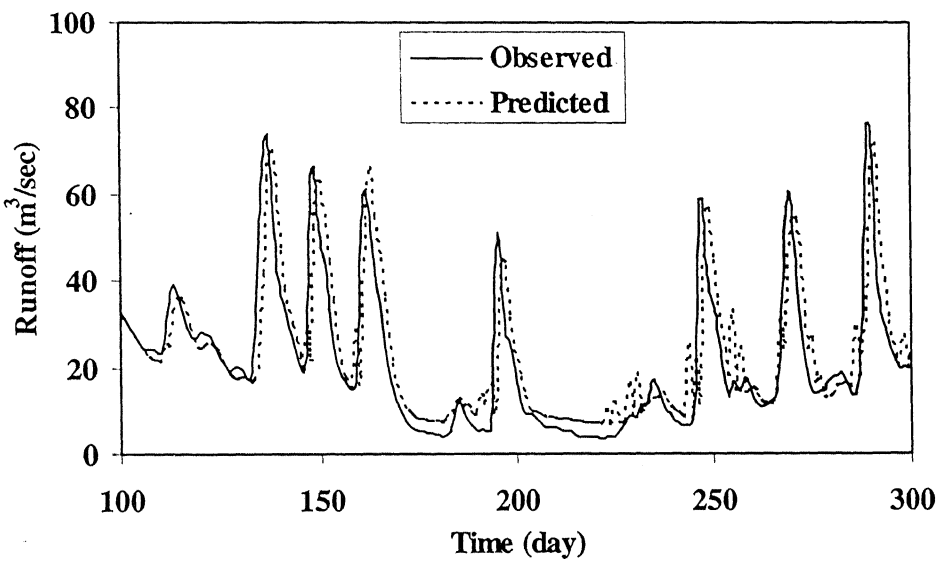


(b) From GANN-GA Model

Figure 5.16: Observed and Predicted Flow in the Wet Year 1989



(b) From GANN-BP Model



(b) From GANN-GA Model

Figure 5.17: Observed and Predicted Low Flows in 1986

distribution can be obtained from GA trained *grey–box* model in all the ranges of flow for both training and testing data sets. The GANN-GA model performs significantly better than the GANN-BP model in predicting the lower magnitude flows as evident from the TS and the AARE statistic. However, in other ranges it gives marginally better or equal estimation capability with GANN-BP model. Similarly, Table 5.3 and Table 5.4 shows significant improvements in R, E, NMBE, and NRMSE statistics also for all ranges of the flow from GANN-GA model for both training and testing period. In summary, it can be said that, like BANN models, the GANN model trained using *elitist* RGA was able to achieve much better generalized relationship of the complex, dynamic, non-linear, and fragmented rainfall-runoff process as compared to the GANN model trained using popular BP algorithm.

5.3.3.2 Comparison of BANN and GANN models

Comparing the results of BANN and GANN models from Table 5.1 during training, it can be observed that the performance of the GANN models is better than the BANN models in efficiency in modeling the complex, dynamic, and non-linear rainfall-runoff process in terms of the R, E, NRMSE and E_{per} statistics. For example, considering the error statistics during training, the value of E_{per} statistic from the BANN-BP model of 0.731 improves to 0.736 from the GANN-BP model; and that of 0.728 from the BANN-GA model improves to 0.735 from the GANN-GA model. Similar improvements can be noticed in terms of R, E, NRMSE and E_{per} statistics during testing. Further, comparing the results of BANN and GANN models from Table 5.1 in effectiveness in predicting the flows accurately, it can be observed that the performance of the GANN models is superior to the BANN models in terms of the TS and AARE statistics during both training and testing. For example, considering the error statistics during training,

the value of AARE from the BANN-BP model of 54.45% improves to 52.20% from the GANN-BP model; and that of 22.39% from the BANN-GA model improves to 21.68% from the GANN-GA model. Similar improvements can be noticed from Table 5.1 in terms of TS and AARE statistics during testing data set. The results of the partitioning analysis, presented in Table 5.3 and Table 5.4 for training and testing data sets, respectively, also indicate that the performance of the GANN models is marginally better than that of the BANN models in modeling the low, medium, and high magnitude flows. However, one interesting observation was that the BANN-GA model was able to predict the maximum flow value with the least error, as evident from the %MF statistics, among all the models considered in this section.

Further, a close examination of the scatter plots of the observed and predicted flows for the validation data set of 13-years from the BANN models (Figure 5.6) and the GANN models (Figure 5.14) indicates that scatter points lie around the 45-degree line in a narrower band from GANN models as compared to that from the BANN models. Further, Figures 5.9 and 5.17 indicate that the estimation of low magnitude flows is much better from the GANN models as compared to the BANN models. Further, considering all the eight performance evaluation indices measuring the efficiency in modeling and effectiveness in predicting the flows and the graphical results together, it can be said that the *grey-box* ANN model trained using *elitist* RGA (GANN-GA model) was the best among the four ANN models under investigation in this section of the thesis. Therefore, considering the CRR, BANN, and GANN models together, it can be said that the GANN-GA model is the best among all the models discussed so far in this study for the purpose of modeling the complex, dynamic, and non-linear rainfall-runoff process. In order to further improve upon the efficiency and effectiveness of the GANN

models, decomposition technique was explored for modeling the complex, dynamic, non-linear, and fragmented rainfall-runoff process for the data considered in this study.

5.4 *Gray-Box* Decomposed ANN (GDANN) Models

The rainfall-runoff mapping in a watershed can be fragmented or discontinuous having significant variations over the input space because the functional relationships between rainfall and runoff being quite different for low, medium, and high magnitudes of streamflow (Zhang and Govindaraju, 2000). Many researchers have experienced difficulties in developing ANN rainfall-runoff models using the training data consisting of whole range of flow magnitudes (Hsu et al. 1995; Tokar and Markus 2000). The ASCE task committee report (2000 a, b) stressed that the future efforts should be directed towards designing ANNs in order to capture the rainfall-runoff relationships inherent in both normal and extreme conditions. There can be several ways in which the rainfall-runoff relationships for low, medium, and high magnitude flows can be developed. One method can be first dividing the data into the different categories depending upon the relative magnitudes of flows. Such type of methods is based on experience and may be called heuristic methods. Zhang and Govindaraju (2000) used such a heuristic approach in first decomposing the data into low, medium, and high magnitude categories and then developing separate ANN models for each category, which were then embedded into a single modular ANN. Another approach is to decompose the input output data using soft computing techniques, such as self-organizing maps (SOM). Furundzic (2000) employed SOM networks to decompose the data into different classes before developing BP trained ANN models for each class. Apart from these two studies, the efforts in the area of decomposing the big problem of modeling the complex, non-linear, and complex rainfall-runoff process into a number of

smaller simpler pieces are lacking. Moreover, none of these two approaches uses the physics of the hydrological process to decompose the input and output data. An approach that decomposes the problem into a few smaller ones based on the physical processes will be more desirable and reliable. The present study makes an attempt to decompose the rainfall and runoff data into different classes based on the different dynamics inherent in the different segments of a flow hydrograph.

The objective of the development of mathematical models of the complex, dynamic, non-linear and fragmented rainfall-runoff process for operational hydrology is to predict stream flow on a continuous basis. A streamflow sequence to be predicted can be considered to be composed of many flow hydrographs that are produced when time varying input impulses in the form of rainfalls is applied to a watershed. It is to be noted that the shape of a flow hydrograph is representative of the combination of multitude of factors such as watershed and storm characteristics including storage, infiltration, and soil moisture conditions. When an ANN rainfall-runoff model is developed using a continuous record of the rainfall and runoff data, the network essentially attempts to learn the complex and dynamic nature of the physical processes of the transformation of rainfall into flow hydrograph during the training phase. Most of the ANN applications reported in literature attempt to model the complex, dynamic, non-linear, and fragmented rainfall-runoff process represented in a flow hydrograph, using a single ANN. However, it must be realized that the runoff response of a watershed, represented by different segments of a flow hydrograph, is produced by different physical processes ongoing in a watershed. For example, the rising limb of a flow hydrograph is the result of the gradual release of water from various storage elements of a watershed (e.g. surface and sub-surface storage) due to gradual repletion of the storages due to the

rainfall input. The rising limb of the hydrograph is influenced by varying infiltration capacities, watershed storage characteristics, and the nature of the input i.e. intensity and duration of the rainfall, and not so much by the climatic factors such as temperature and evapotranspiration, etc. (Zhang and Govindaraju, 2000). On the other hand, the falling limb of a flow hydrograph is the result of the gradual release of water from the watershed after the rainfall input has stopped, and is influenced more by the storage characteristics of the watershed and climatic characteristics to some extent. Therefore, the use a single ANN to represent the input output mapping of the whole hydrograph may not be as efficient and effective as compared to developing two different mappings representing the two limbs of the flow hydrograph. In fact, one can go a step further and note that the physical processes in the watershed responsible in producing the initial portions of the rising limb (R1) are different than the physical processes responsible in producing the latter portions of the rising limb close to the peak discharge (R2). This is because the soil moisture and watershed storage conditions are on the drier side in the beginning of a storm, and larger infiltration rates dominate the flatter portions of the flow hydrograph. On the other hand, the soil moisture and watershed storage conditions are close to saturation during the middle of the storm close to peak, and the lower infiltration rates dominate the steep latter portions of the rising limb of a flow hydrograph. Similarly, the same may hold true for the falling limb of the hydrograph where initial portions (F1) are more influenced by interflow, middle portions are dominated by the delayed surface flow (F2), and the lower portions (F3) may be dominated by base flow. This is explained in Figure 5.18, which shows a decomposed flow hydrograph.

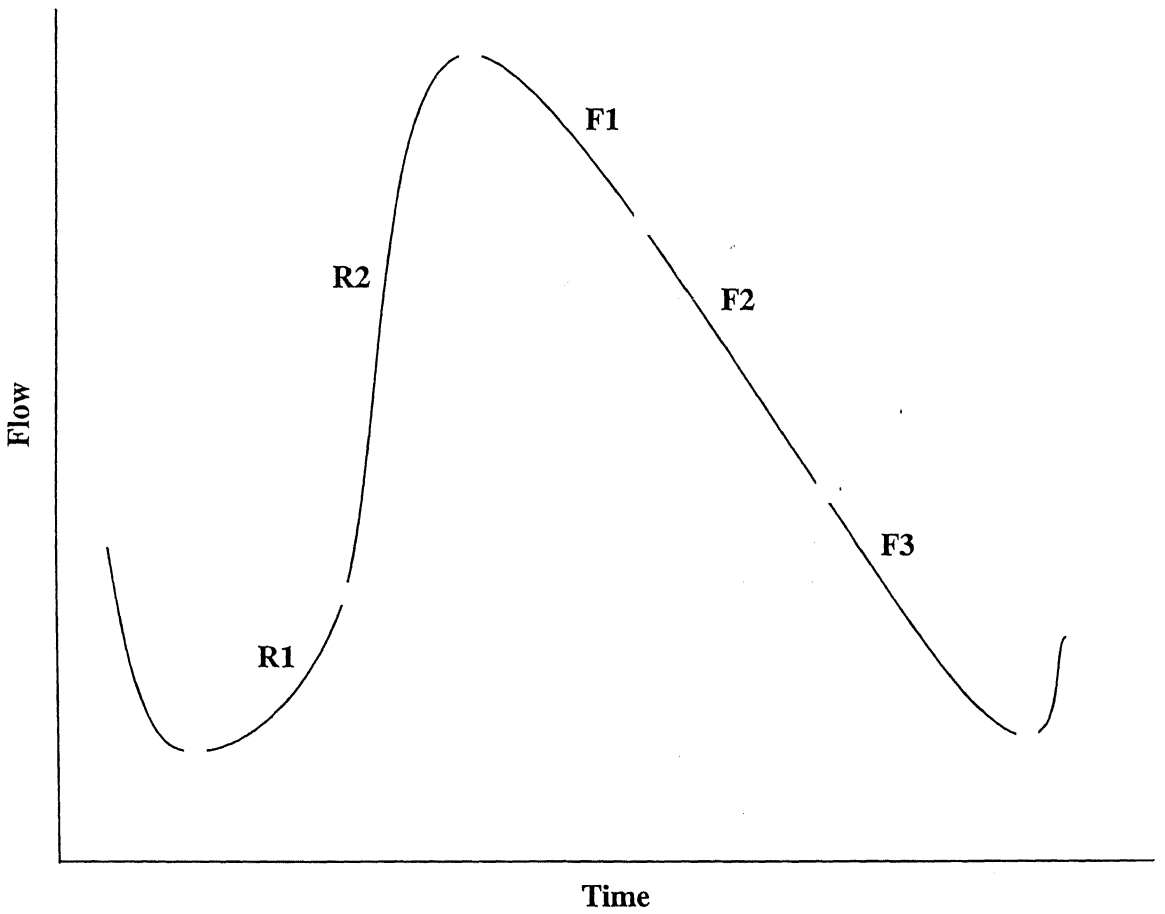


Figure 5.18: Decomposition of a Flow Hydrograph

Therefore, it will be appropriate to decompose the data corresponding to different segments of a flow hydrograph into different categories, and then attempt to model the rainfall-runoff relationships in those categories. This study gradually decomposes the data into increasing number of classes in order to develop a separate category of models called "*grey-box* ANN models of the decomposed flow hydrograph" (GDANN models). Seven different types of GDANN model structures have been investigated in this study that differ in (a) varying degree of conceptualism embedded in the ANN models, (b) manner in which different segments of a flow hydrograph are modeled, (c) training methodology used, and (d) the method employed for decomposition.

The first model (GDANN-I) decomposes the flow hydrograph into two parts, a rising limb and a falling limb, and then models each of them using two separate ANNs. The second model (GDANN-II) is same as GDANN-I on the rising limb but models the falling limb using the concept of flow recession employed in the CRR models described earlier in Chapter 4. The GDANN-II model was developed to assess the relative performance of the ANN and the conceptual technique of flow recession in modeling the falling limb of a flow hydrograph. The third model (GDANN-III) is the same as GDANN-II on the rising limb, and further decomposes the falling limb into two parts. The first part of the falling limb after the peak, which is dominated by interflow and delayed surface flow (F1+F2 in Figure 5.18), is modeled using the ANN technique, and the second part, which is dominated by the base flow (F3 in Figure 5.18), is modeled using the conceptual technique. The fourth model (GDANN-IV) is same as GDANN-III on the falling limb, and decomposes the rising limb also into two parts. The first part of the rising limb consisting of the initial portions (R1), which is characterized by higher infiltration capacities and drier watershed storage conditions, is modeled using a

conceptual technique, and the second part consisting of the latter portions of the rising limb (R2), which is characterized by soil moisture and watershed conditions close to saturation, is modeled by the ANN technique. The first four models (GDANN-I through GDANN-IV) mentioned above decompose the data using heuristic technique based on the physics of the hydrological processes, and trained the ANN models using BP method. The fifth model structure developed in this category (GDANN-V) was same as GDANN-IV model except that the ANN components in the GDANN-V model were trained using *elitist* RGA.

Another type of decomposition technique employed in the present research effort was the soft computing technique of SOM networks. Two different SOM networks were developed which divided the input output data into three and four classes, respectively. These are designated as SOM (3) and SOM (4) models in the thesis. After classifying the data into three and four classes using SOM classifiers, individual ANN models were developed in each class for both the models. The development of the GDANN models is discussed next.

5.4.1 GDANN Models based on Heuristic Decomposition

All the ANN models developed in this section consist of three layers; one input layer, one hidden layer, and one output layer. The task of identifying the number of neurons in the input and output layers is simple as it is dictated by the input and output variables involved in the problem. In this study, GANN-BP model which was developed in previous section is taken as the base-model consists of the input variables of present and past *effective* rainfalls, ER(t), ER(t-1), and ER(t-2), computed using the Green-Ampt equations and past flow values, Q(t-1) and Q(t-2), while the only neuron in the output

layer represented the flow at time t, Q(t), being modeled. The same input and output training patterns are used to develop remaining structures of individual ANN models, GDANN-I through GDANN-V. The input-output pairs are disintegrated as rising and falling limb using the previous three runoff values and previous day effective rainfall information. The input vector for the independent ANN models were determined for the GDANN-I through GDANN-V developed above using the cross correlation analysis between corresponding input output data. The optimum architecture of independent ANN models of the GDANN-I through GDANN-V, and the number of patterns used to train them, and the associated input variables are presented in Table 5.5. The number of neurons in the hidden layer has to be determined using corresponding training data through the use of a trial and error procedure. The value of learning coefficient of 0.01 and momentum correction factor of 0.075 was used while training the all the networks developed in this study. The GA parameters used for the GANN-V model were same that for the GA trained models developed in the sections 5.2 and 5.3.

The division of both falling and rising limbs for the GDANN-II and GDANN-III models was achieved through the use of three error statistics corresponding to different values of dividing the flow. The error statistics were NRMSE, NMBE (%) and t-statistic. The NRMSE and NMBE (%) statistics were described in Chapter 4, and the t-statistics use the Mean Bias Error (MBE) and Root Mean Square Error (RMSE) between observed and predicted streamflow values. The smaller the value of the *t*-statistic, the better is the model's performance. The t-statistic, as proposed by Stone (1993) can be computed using the following equation:

$$t = \left(\frac{(N-1)MBE^2}{RMSE^2 - MBE^2} \right)^{1/2} \tag{5.2}$$

Figure 5.19 and Figure 5.20 show the plots of various error statistics for different levels of flow values for dividing the falling limb and rising limb, respectively. It is clear from Figure 5.19 that a flow value of 70.8 m^3/s (or 2500 ft^3/s) is most suitable for the purpose of dividing the falling limb as most of the error curves converge to a minimum at this flow value. Similarly, Figure 5.20 indicates that a flow value of 11.33 m^3/s (400 ft^3/s) is suitable for dividing the data on the rising limb, as the t-statistics and NMBE were minimum at this flow value, and the variation in NRMSE is not much. Once the critical flow values on both rising and falling limbs were determined, the data were divided into different classes as depending on the magnitude of the flow value on rising and falling limbs. Then, the data in each class were further broken into training and testing data sets for ANN model building. Then the corresponding training data sets were used to train the respective ANN models for the corresponding portions of the falling or rising limb, in models GDANN-I through GDANN-V discussed earlier. Once the training of the ANN models was complete, they were validated using the appropriate testing data set. The flow chart for the methodology for prediction from the most complex GDANN-IV model is given in Figure 5.21.

5.4.2 GDANN Models based on Soft Decomposition

Another type of ANN models developed in this study is self-organizing map (SOM) models, which employ unsupervised training algorithm to divide data into different classes corresponding to different dynamics. The input data utilized to develop these ANN models are same as the base model (GANN-BP). Two different SOM models were developed: the first one, called SOM(3), explored the possibility of three classes in the rainfall-runoff mapping, and the second one, called SOM(4), explored for the possibility of four classes in the rainfall-runoff mapping. The details of SOM networks

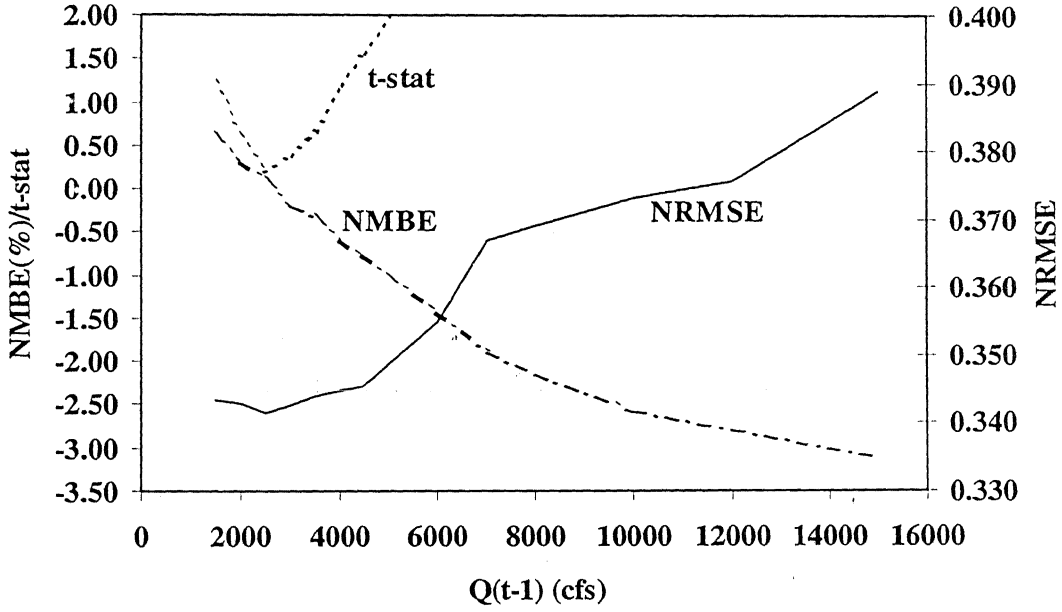


Figure 5.19: Dividing the Falling Limb

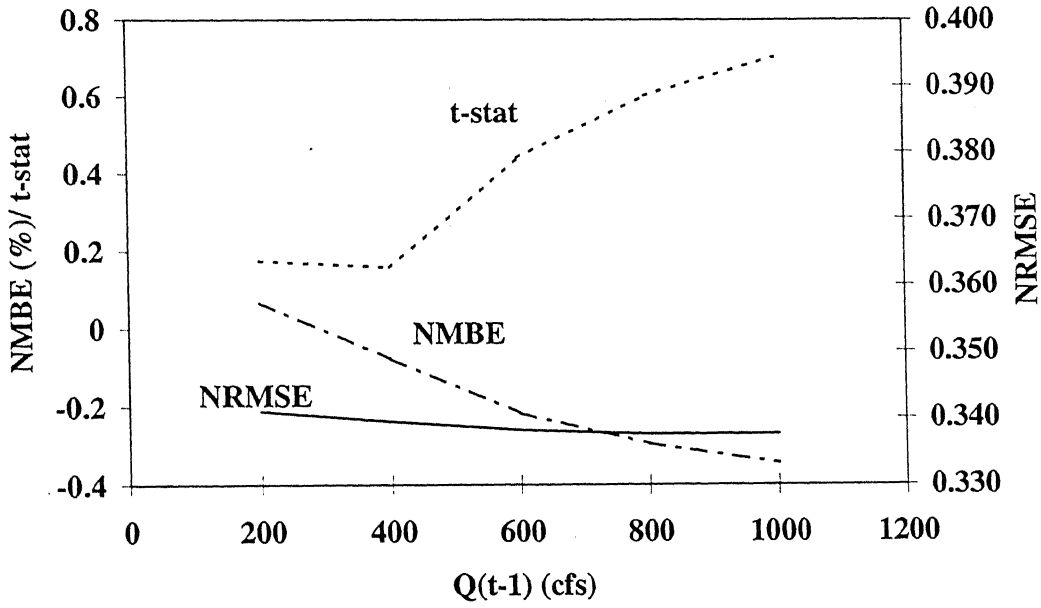


Figure 5.20: Dividing the Rising Limb

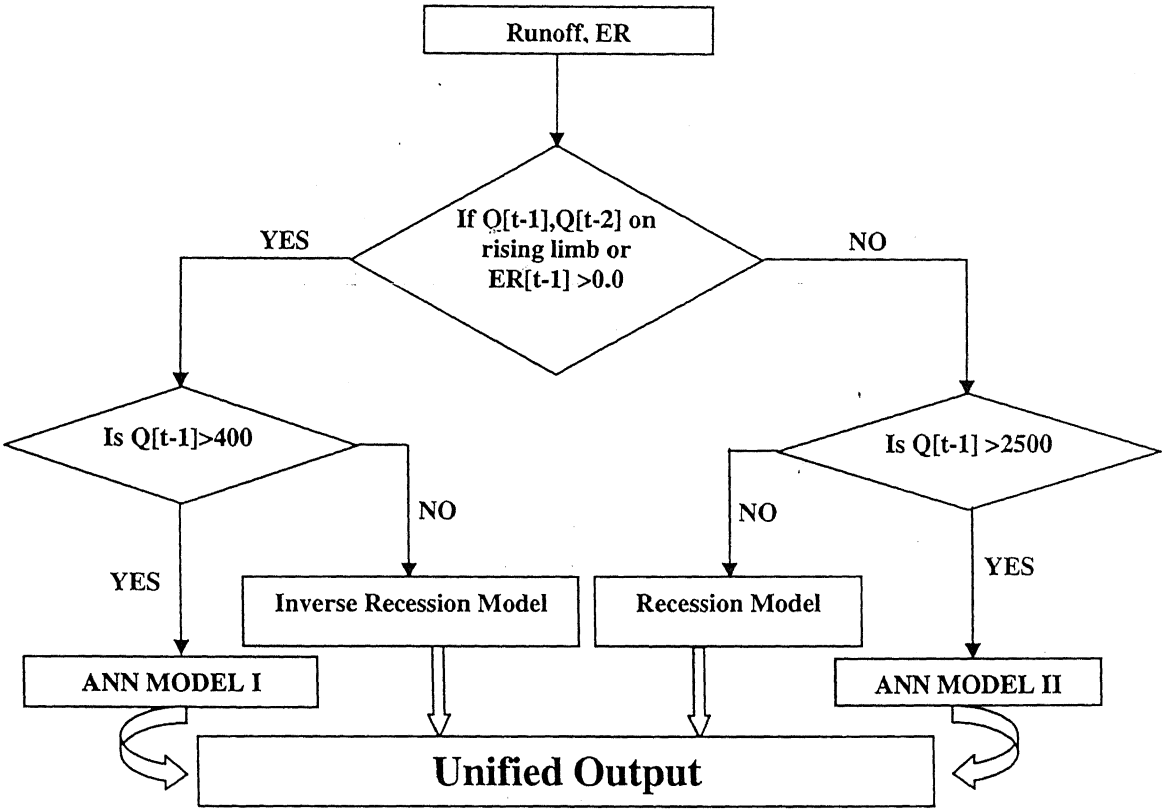


Figure 5.21: Flow Chart for GDANN-IV Model

Table 5.5: Details of GDANN Model Structures

Model	Portion	Architecture	Number of Data	Input Variables
GANN-BP		5-4-1	4747	ER(t), ER(t-1), ER(t-2), Q(t-1), and Q(t-2)
GDANN-I	Rising	5-4-1	1783	ER(t), ER(t-1), ER(t-2), Q(t-1), and Q(t-2)
	Falling	3-3-1	2963	ER(t), Q(t-1), and Q(t-2)
GDANN-II	Rising	5-4-1	1783	ER(t), ER(t-1), ER(t-2), Q(t-1), and Q(t-2)
	Falling	Recession	2963	Q(t-1) and Q(t-2)
GDANN-III	Rising	5-4-1	1783	ER(t), ER(t-1), ER(t-2), Q(t-1), and Q(t-2)
	Falling-I	3-3-1	1189	ER(t), Q(t-1), and Q(t-2)
	Falling-II	Recession	1774	Q(t-1) and Q(t-2)
GDANN-IV	Rising-I	Inverse Recession	182	Q(t-1) and Q(t-2)
	Rising-II	5-4-1	1601	ER(t), ER(t-1), ER(t-2), Q(t-1), and Q(t-2)
	Falling-I	3-3-1	1189	ER(t), Q(t-1), and Q(t-2)
	Falling-II	Recession	1774	Q(t-1) and Q(t-2)
GDANN-V	SAME			
SOM(3)	Class-I	5-4-1	693	ER(t), ER(t-1), ER(t-2), Q(t-1), and Q(t-2)
	Class-II	4-3-1	2993	ER(t), ER(t-1), Q(t-1), and Q(t-2)
	Class-III	3-3-1	1061	ER(t), Q(t-1), and Q(t-2)
SOM(4)	Class-I	5-4-1	409	ER(t), ER(t-1), ER(t-2), Q(t-1), and Q(t-2)
	Class-II	4-3-1	704	ER(t), ER(t-1), Q(t-1), and Q(t-2)
	Class-III	3-3-1	1089	ER(t), Q(t-1), and Q(t-2)
	Class-IV	3-3-1	2545	ER(t), Q(t-1), and Q(t-2)

and their training were discussed in detail in Chapter 3. Once the input space was divided into different classes using the SOM classifier, the data from each class were used to develop independent ANN models using supervised training method as discussed earlier. The input architecture of the individual ANN model was decided using cross-correlation analysis between input and output data sets of the corresponding class of observations from training data set. The optimum number of hidden neurons is determined using the trial and error method. The optimum architecture of independent ANN model for the corresponding class of observations, and the number of patterns used to train, and the associated input variables are presented in Table 5.5. Once the model structure was identified, it was used to produce unified output in terms of estimated streamflow from the model for both training and testing data sets. The flow chart for computing streamflow from SOM (4) is depicted in Figure 5.22.

5.4.3 Discussion of Results from GDANN Models

The results obtained from base model, GDANN-I model through GDANN-V model, SOM (3), and SOM (4) models in terms of various performance evaluation indices during training and testing data sets are presented in Table 5.6 and Table 5.7, respectively.

It can be noticed from Table 5.6 that the performance of various GDANN models becomes better as we move down the table from BANN-BP to GDANN-V model. The GDANN-V performs the best in terms of most of the TS statistics, AARE, correlation coefficient, efficiency, NRMSE, %MF, and E_{per} and comparable in terms of NMBE (%) statistics. The best values of AARE, R, E, NRMSE, and E_{per} statistics of 16.47%, 0.9785, 0.9575, 0.335, and 0.749, respectively, were obtained from GDANN-V

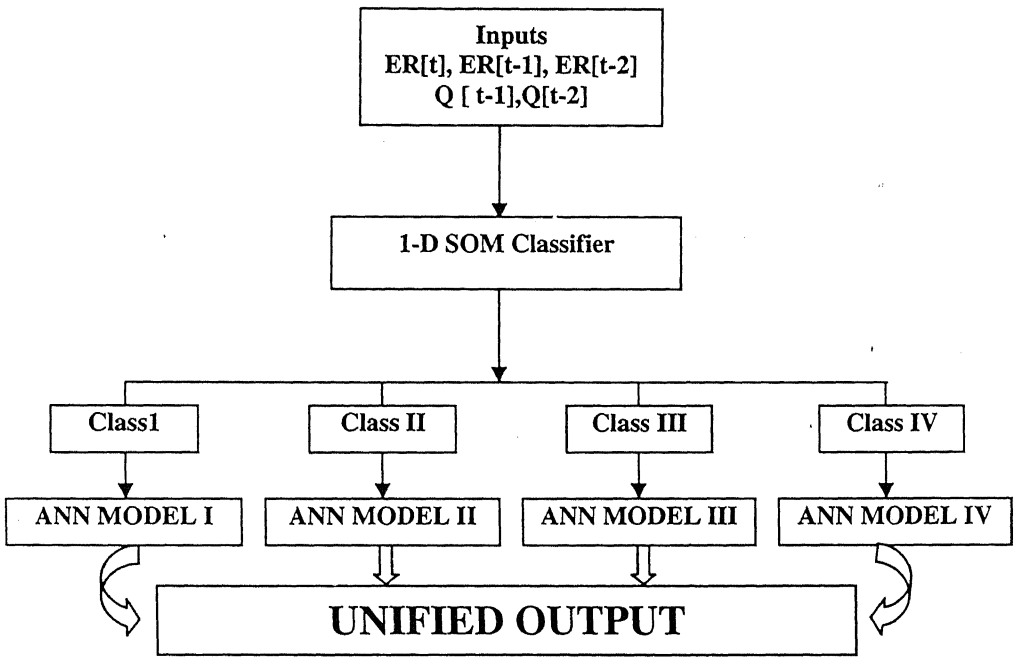


Figure 5.22: Flow Chart for SOM(4) Model

Table 5.6: Performance Evaluation Indices from GDANN Models during Training

Model	TS1	TS5	TS25	TS50	TS100	AARE	R	E	NMBE(%)	NRMSE	%MF	E_{per}
GANN-BP	3.20	14.17	51.67	69.37	82.19	54.97	0.9770	0.9545	1.68	0.347	-4.53	0.736
GDANN-I	2.63	12.92	50.29	67.45	80.17	61.28	0.9764	0.9531	1.87	0.353	-5.03	0.722
GDANN-II	6.76	27.79	72.19	85.00	92.65	31.66	0.9607	0.9215	-1.75	0.456	-5.03	0.536
GDANN-III	6.11	25.20	71.87	85.55	92.60	31.90	0.9777	0.9560	0.10	0.342	-5.03	0.740
GDANN-IV	6.25	26.70	74.71	88.55	95.40	23.85	0.9780	0.9570	-0.06	0.339	-5.10	0.743
GDANN-V	6.31	28.71	79.73	94.22	99.24	16.47	0.9785	0.9575	-0.76	0.335	-2.30	0.749
SOM(3)	4.63	21.36	70.00	88.39	97.85	23.22	0.9793	0.9591	0.21	0.329	-5.81	0.759
SOM(4)	4.27	22.06	72.27	90.89	98.44	20.80	0.9804	0.9612	0.12	0.320	-6.57	0.772

Table 5.7: Performance Evaluation Indices from GDANN Models during Testing

Model	TS1	TS5	TS25	TS50	TS100	AARE	R	E	NMBE(%)	NRMSE	%MF	E_{per}
GANN-BP	2.94	14.51	54.93	69.80	80.15	65.71	0.9700	0.9406	-0.19	0.389	-2.89	0.689
GDANN-I	2.85	12.98	52.96	68.98	79.22	72.28	0.9696	0.9398	-0.02	0.393	-3.05	0.681
GDANN-II	7.73	31.23	73.02	85.48	92.16	36.45	0.9571	0.9150	-2.44	0.466	-3.05	0.549
GDANN-III	6.47	25.67	67.29	83.44	91.80	39.56	0.9689	0.9376	2.22	0.398	-3.05	0.674
GDANN-IV	7.01	28.08	72.37	89.62	97.62	21.63	0.9678	0.9388	0.81	0.394	-3.22	0.684
GDANN-V	7.12	28.22	76.14	93.13	99.28	17.96	0.9686	0.9389	0.26	0.397	-0.76	0.682
SOM(3)	4.29	20.75	63.06	83.43	94.81	28.59	0.9680	0.9367	-2.24	0.401	-3.65	0.665
SOM(4)	2.59	13.73	61.75	86.17	97.61	26.51	0.9620	0.9100	9.63	0.478	-4.08	0.523

model. Also, more than 99% of the estimated flow values from GDANN-V model had absolute relative error (ARE) less than 100%. Hence, it can be said that the model that employed integration of conceptual and ANN techniques, used *elitist* RGA to train ANN models, and modeled different parts of the flow hydrograph using different technique performed the best during training among the GDANN models investigated in this section of the thesis. Further, the performance of both the SOM models is better than GDANN-V in terms of R, E, NMBE(%), NRMSE, and E_{per} but worse in terms of TS and AARE statistics in the training phase. Further, it can be noted that both the SOM models perform better than GDANN-I through Model-IV in terms of most of the performance evaluation statistics suggesting that an approach, which models the rainfall-runoff process by decomposing the input space into different classes corresponding to different dynamics of rainfall-runoff process, is better than a lumped approach of considering whole input space as a single unit for the purpose of rainfall-runoff modeling. Further, within the SOM models, the SOM(4) model performed better than the SOM(3) model in training data indicating that the rainfall-runoff data from Kentucky River watershed possibly consists of four classes corresponding to different dynamics. This finding is in line with dividing the flow hydrograph into four parts corresponding to different dynamics in rainfall-runoff mappings while developing the GDANN models.

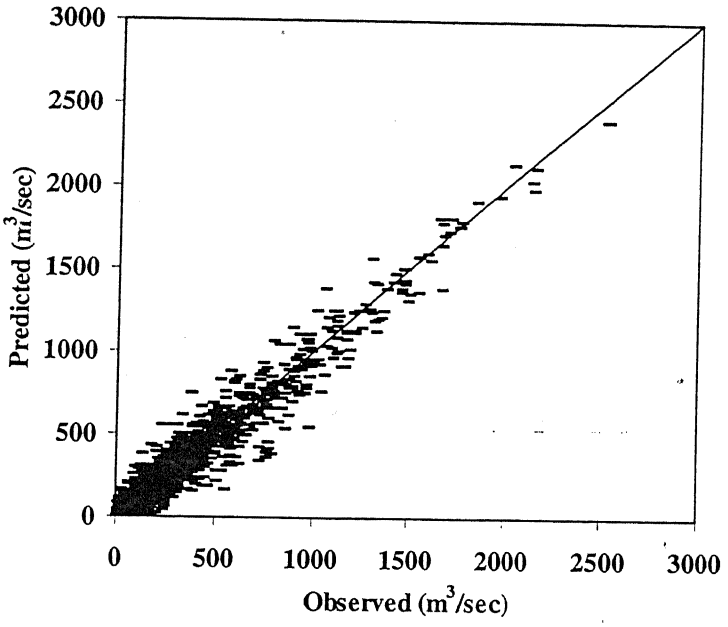
Observing the performance of various models during testing, it can be noted that the GDANN-V was superior to the remaining GDANN models in terms of effectiveness in predicting flow values accurately as can be seen from all TS and AARE statistics in Table 5.7. For example, the best AARE value of 17.96% was obtained from GDANN-V model. In more than 99% of the predicted cases, the GDANN-V model had ARE

values less than 100%. The GANN-BP was found to be better in terms of R, E, and NRMSE statistics, and GDANN-I model was the best in terms of NMBE statistic during testing though marginally. However, the capability of these models in effectively predicting flow accurately was very poor (see TS and AARE in Table 5.7). Further, the efficiency of GDANN-V model in modeling the rainfall-runoff process in terms of R, E, NMBE(%),NRMSE, and E_{per} was comparable to the best statistics from other models. It is apparent that the SOM models do not perform as well during testing as they did during training. This may be because of limiting capabilities of input space decomposition algorithm and mapping algorithm that results on testing data and reduces the effectiveness of the models. Further, the SOM (4) model performed better then SOM (3) model in all the statistical criteria except the %MF statistics in training. But it does not do well during testing data set in terms of all error statistics, compared to SOM(3) and GDANN-V models due to because of the same reason explained above. Further, in our case it must be realized that increasing number of classes using the SOM classifiers is biased towards the class belongs to more number of data points to classify the data set and become less effective in classifying the patterns reducing the efficiency of the algorithm. The effectiveness of GDANN-V model indicates that the heuristic delineation of boundaries between different physical processes performs better than the soft decomposition methodologies. Therefore the *elitist* RGA trained GDANN model is gave best results both in terms of efficiency and effectiveness for the training and testing data set over all the models developed in this section.

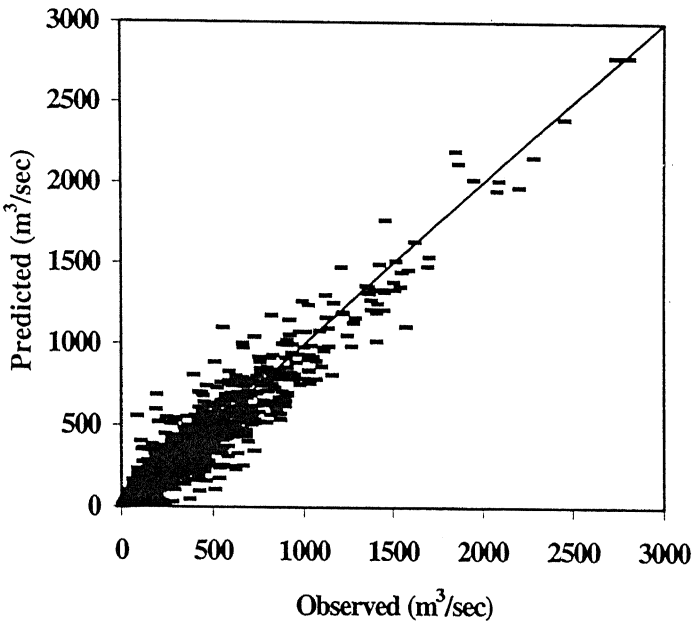
A model that is both efficient in modeling and effective in predicting accurately is more desirable than a model being the most efficient in modeling and not effective in predicting accurately. There seems to be a trade off among the two different

capabilities of a model, therefore, the final model structure needs to be selected based on the optimum trade off desired depending on the application of the model. For example, a daily flow forecasting model is suitable for operational purposes for managing various water resources projects, therefore, a model with good efficiency in modeling and better effectiveness in predicting flow accurately is desirable. Considering these issues, GDANN-V model seems to be the best model trained with *real-coded* GA algorithm for forecasting daily flow in the Kentucky River watershed. The performance of GDANN-IV model and GDANN-V model during testing in graphical form in terms of observed and predicted flow as scatter plots is shown in Figure 5.23. The observed and predicted flow for the dry and wet years 1986 and 1989, respectively, from GDANN-IV model and GDANN-V model are shown in Figure 5.24 and Figure 5.25, respectively. These figures confirm that GDANN-V model is able to predict flow values very accurately.

Further, comparing the performances of GDANN-I model and GDANN-II model, it was found that modeling the falling limb using the concept of flow recession is a better approach than modeling the falling limb using an ANN technique in terms of effectiveness in prediction but not necessarily in terms of efficiency in modeling. That is, ANN technique is good in efficient modeling while the conceptual technique of flow recession is good in predicting flow values more accurately. Therefore, an approach that is based on a combination of these two techniques for modeling falling limb may yield better model performance. This is verified by comparing the results obtained from GDANN-I model and GDANN-II model that are same as rising limb but differ in the manner of modeling the falling limb. The model that decomposed the falling limb into two parts, and modeled initial portions dominated by interflow and delayed flow using

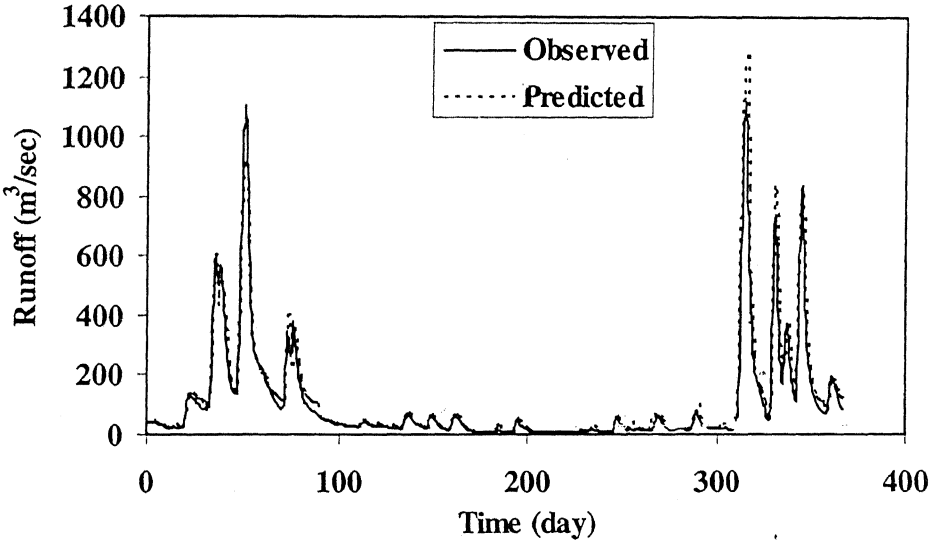


(a) From GDANN-IV Model

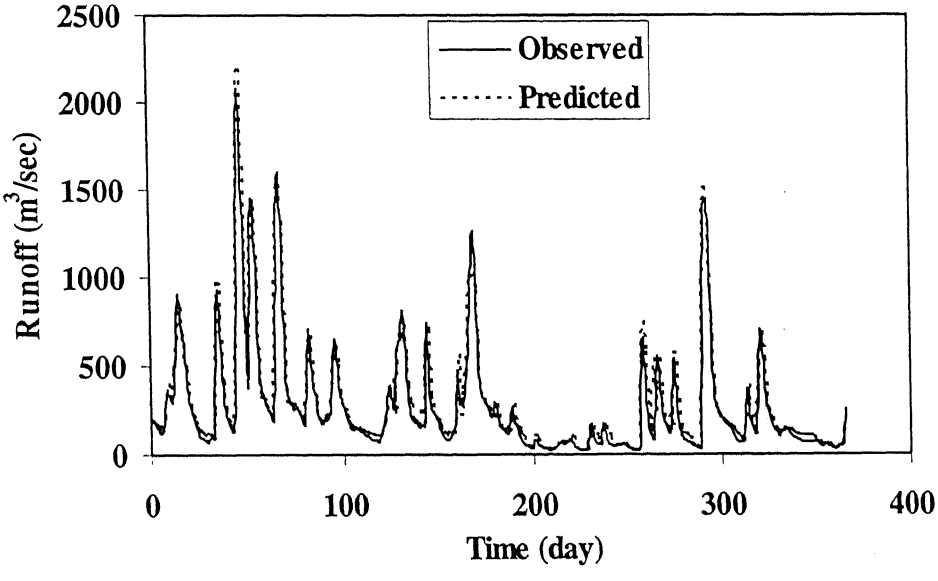


(b) From GDANN-V Model

Figure 5.23: Observed and Predicted Flows during Testing

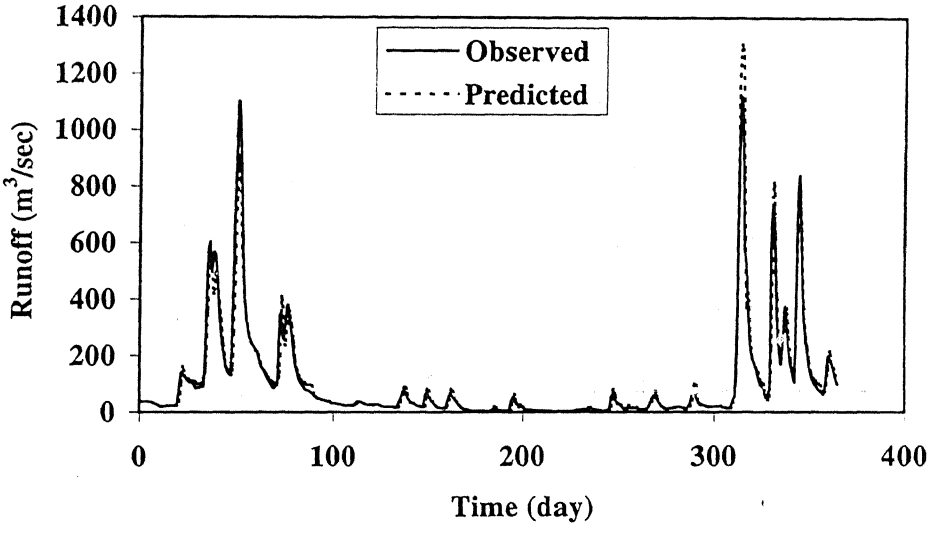


(a) From the Year 1986

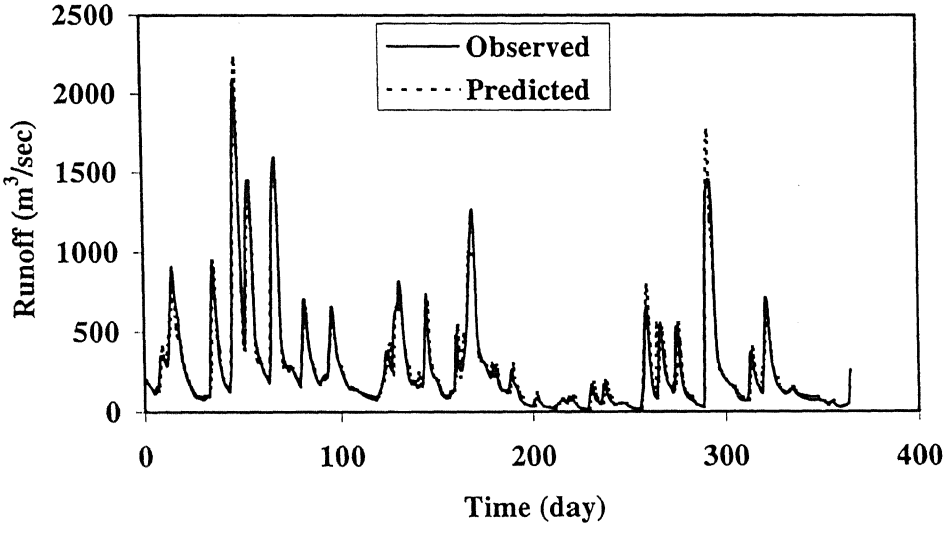


(b) From the Year 1989

Figure 5.24: Observed and Predicted Flows from the GDANN-IV Model



(a) From the Year 1986



(b) From the Year 1989

Figure 5.25: Observed and Predicted Flow from GDANN-V

ANN technique and the latter portions dominated by base flow using conceptual technique (GDANN-III model) was able to achieve good efficiency in modeling (advantage of GDANN-I model) and good effectiveness in prediction (advantage of GDANN-II model). Therefore, it seems that by decomposing the falling limb of the flow hydrograph into two parts and employing an integrated approach of using different techniques to model different portions is a better approach than using a single technique to model the whole falling limb. Further, comparing the performance of GDANN-III model and GDANN-IV model, which are same on the falling limb but differ in the manner of modeling the rising limb, it can be noted that GDANN-IV model performs better than GDANN-III in terms of most of the performance evaluation indices during both training and testing data sets. This suggests that decomposing the rising limb into two portions and employing an integrated approach to model the initial portions of the rising limb, which is dominated by infiltration capacities and drier soil moisture and watershed storage conditions, using a conceptual technique, and latter portions of the rising limb characterized by soil moisture and watershed storage conditions close to saturation using an ANN technique, is a better approach than using a single technique to model the whole rising limb of a flow hydrograph. Further, it was found that the decomposition based on physical processes not only reduces complexity of the relationship but also reduces the training time of the *elitist* RGA.

These findings are indeed promising as they can potentially open up a new dimension for the researchers to explore in order to develop efficient and effective models of the hydrologic process. The concepatul rainfall-runoff models, in fact, use different techniques to model different portions of a flow hydrograph but the efforts of using such an integrated approach in employing ANN technique to hydrologic modeling are

lacking. ANNs are powerful tools of modeling and forecasting non-linear systems such as a flow hydrograph. Therefore, an integrated approach of using ANN and conceptual techniques coupled with decomposition of a larger problem into smaller simpler ones can be quite effective and efficient and needs to be explored further.

5.5 Summary

This chapter describes the improved methodologies for the development of models of the complex, dynamic, non-linear, and fragmented rainfall-runoff process in a large watershed using soft computing techniques, and a combination of the conceptual and soft computing techniques. The techniques investigated include conceptual modeling techniques, ANN technique, *elitist real-coded* genetic algorithm (*elitist* RGA), and physics based heuristic and decomposition techniques. Three types of rainfall-runoff models were developed: (a) black box type ANN rainfall-runoff models called BANN models, (b) *grey-box* type ANN rainfall-runoff models that consider the physics of the hydrologic process in a partial sense, called GANN models, and (c) *grey-box* ANN rainfall-runoff models using decomposition of the flow hydrograph, called GDANN models. Specifically, two BANN models, two GANN models, and seven GDANN models were developed. The BANN and GANN models were trained using both popular back-propagation (BP) algorithm and *elitist* RGA. Among the seven GDANN models, the first five models (GDANN-I model through GDANN-V model) used heuristic decomposition based on the dynamical processes inherent in different segments of a flow hydrograph, and the last two models (SOM(3) and SOM(4) models) used soft decomposition technique of the self organizing map (SOM) network classifiers. The first GDANN model divided the rising and falling limbs of the flow hydrographs into two parts and separate ANN models were developed for rising and

falling limbs, respectively. The second GDANN model was same as the first one on the rising limb but modeled the falling limb using the concept of flow recession. The third GDANN model was same as the second one on the rising limb, but further decomposed the falling limb into two parts. The first part of the falling limb after the peak, which is dominated by interflow and delayed surface flow, was modeled using the ANN technique, and the second part, which is dominated by the base flow, was modeled using a conceptual technique. The fourth GDANN model was same as the third GDANN model on the falling limb, and decomposed the rising limb into two parts. The first part of the rising limb consisting of the initial portions, which is dominated by higher infiltration capacities and drier soil moisture and watershed storage conditions, was modeled using a conceptual technique, and the second part consisting of the latter portions of the rising limb characterized by soil moisture and watershed storage conditions close to saturation, was modeled by the ANN technique. The fifth GDANN model was same as the fourth one but the ANN components were trained using *elitist* RGA. All the GDANN models employed effective rainfall as inputs in the ANN model component, which was computed by first estimating the infiltration using Green-Ampt equations. The SOM(3) and SOM(4) models divided the input output data into three and four classes, respectively, and separate ANN models were developed for the divided classes of data. The daily rainfall and streamflow data derived from the Kentucky River watershed, USA were employed to train and test all the BANN, GANN, and GDANN models investigated in this chapter. Eight different performance evaluation indices were computed from all the models both during training and testing data sets in order to evaluate their relative strengths and weaknesses. The results in terms of various performance evaluation indices from the BANN, GANN, GDANN, and SOM models are presented in Table 5.1. The performance of the BANN models was found to be

better than the CRR models in terms of efficiency in modeling but CRR models performed slightly better than the BANN models in terms of effectiveness in prediction. This was due to the poor generalization ability of the popular back-propagation (BP) algorithm employed in the BANN models for training the ANNs, especially the low magnitude flows, which was overcome by the use of *elitist* RGA in training the BANN models. The results obtained in this study indicate that the incorporation of conceptual components in the ANN models improves the overall efficiency and effectiveness of the ANN rainfall-runoff models marginally. The marginal improvements in GANN models over the BANN models may be attributed to the efficiency of the conceptual techniques themselves or the errors in the rainfall data employed in this study. However, the training of GANN models by the *elitist* RGA also showed significant improvements in the overall generalization capability of the ANNs in modeling the complex, dynamic, non-linear, and fragmented rainfall-runoff process. The gradual decomposition of the flow hydrograph into different segments using the physics based heuristic decomposition and the use of separate techniques to model different segments of the flow hydrograph, was found to result in gradual improvements in both effectiveness in modeling and effectiveness in predicting flows accurately from such models of the complex, dynamic, non-linear, and fragmented rainfall-runoff process. The *grey-box* rainfall-runoff model that employed physics based decomposition technique, and uses *elitist* RGA for training the ANN component (GDANN-V model) was found to be excellent in efficiency in modeling and effectiveness in prediction, and was deemed to be the best model among all the models structures investigated in this study. The decomposition of the input output data into different classes using the SOM coupled with ANN models supports the concept that different dynamical processes inherent in different data sets belonging to different classes should be modeled separately to get

better model performance. Further, the superiority of GDANN-V model over the SOM models indicates that dividing rainfall-runoff data space into different classes corresponding to different dynamics based on conceptual methods is a better approach than the soft decomposition using SOM network classifiers.

Chapter 6

Exploration of Physical Significance in ANN Models

6.1 General

The emergence of artificial neural network (ANN) technology has provided many promising results in the field of hydrology and water resources simulation. However, one of the major criticisms of ANN hydrologic models is that they do not consider/explain the underlying physical processes in a watershed, resulting in them being labeled as black box models. For instance, Lange (1999) mentioned that ANNs are black box models as they do not model any kind of physical processes but only the relation between input and output variables. However, the hydro-meteorological data that are employed in developing rainfall-runoff models (ANN or conceptual) contain important information about the physical process being modeled, which gets embedded or captured inside the model during training/calibration. If the structure of a black-box ANN rainfall-runoff model is employed further, it may be possible to assign physical significance to certain parallel distributed component(s) of the ANN; however, it needs to be investigated. A few studies that have been conducted recently to address this deficiency proposed solutions that represent the operation of a trained ANN in terms of symbolic rules (e.g. Lozowski, et al., 1996; Benitez, et al, 1997; Tickle et al, 1998; Castro et al., 2002). However, extracting the knowledge embedded within trained ANNs is still an active and evolving area of research. Despite these developments and a

plethora of ANN applications in hydrology and water resources, the hydrologists have not endeavored to construe the knowledge embedded in the trained ANN hydrologic models.

In the context of rainfall-runoff modeling, it must be realized that any mathematical model, conceptual or ANN, is built using the rainfall-runoff data derived from a watershed. The overall data framework with respect to space and time contains important information about the complex, non-linear, dynamic, and fragmented rainfall-runoff relationship of the watershed. Such a relationship is normally inherent in the flow hydrographs produced by a sequence of rainfalls occurring at certain time intervals throughout the watershed. Such complex relationships get captured by the mathematical model either in its structure or in the calibrated parameters. The calibrated parameters of a CRR model represent physical characteristics of the watershed, e.g. the value of infiltration parameters, hydraulic conductivity (K), suction head (ψ), and porosity (η) provide useful information about land use and soil conditions of the watershed, which correspond to certain form/type of rainfall-runoff relationship. Similarly, the optimized weights in an ANN and its massively parallel distributed structure capture the physical process(s), being modeled while training of the ANN. The ANNs are much more robust and efficient tools in generalizing rainfall-runoff relationships from a given set of data as compared to CRR models. This aspect of ANN rainfall-runoff models has been demonstrated in many studies (Hsu et al. 1995; Tokar and Johnson 1999). It was mentioned earlier in Chapter 1 and Chapter 5 that different portions of a flow hydrograph have different characteristics and are produced by different physical process ongoing in a watershed. Also, the rainfall-runoff relationships for low, medium, and high magnitude flows are of different types as inherent in different shapes, slopes, and

sizes of a flow hydrograph and associated climatic data. It is possible that the different components of the massively parallel structure of an ANN rainfall-runoff model capture different rainfall-runoff relationships corresponding to different dynamics inherent in low, medium, and high magnitude flows or different portions of a flow hydrograph. Hence, if the components of this distributed structure of an ANN rainfall-runoff are explored further, one may be able to shed some light on the inherent physical processes embedded in the hydrologic black-box models.

However, identifying components of a physical process in a trained ANN is an area of research that is more or less virgin and remains unexplored, especially in hydrology. There are many techniques, e.g. sensitivity analysis and correlation analysis that can be employed to analyze a trained ANN to explore for the possibilities of physics embedded inside an ANN. Owing to the fact that ANN models have large degrees of freedom in assigning the weights, it can lead to a situation where two completely different sets of weights can yield nearly identical outputs (Schmitz et al., 1999). A sensitivity analysis of the parameters of the ANN models may, therefore, lead to insignificant implications. However, the distributed nature of information processing in ANN implies that an ANN can be disaggregated in terms of its forecasting inputs, and still generate outputs. Therefore, it may be possible to explore the physical process being mapped by an ANN by examining the knowledge about a given problem domain in the form of different conceptual components of the physical process along with the input variables, in relation with the components of the distributed structure of an ANN rainfall-runoff model.

This chapter of the thesis presents the results of an explorative study to investigate whether the massively parallel components of a trained ANN hydrological model that are distributed in nature represent the different components of a hydrological process inherent in different portions of a flow hydrograph. In this study, this was achieved by using the developed black-box ANN model (BANN-BP) and a conceptual rainfall-runoff model (CRR Model-II), and then performing the correlation and graphical analyses to decipher any physical processes embedded in the trained ANN rainfall-runoff model with the help of the CRR model components. The rainfall and streamflow data derived from the Kentucky River watershed are employed to investigate the proposed methodology of identifying physical processes inherent in the ANN rainfall-runoff model (BANN-BP). The black-box ANN model trained using BP algorithm was selected because it did not consider the underlying physical processes while developing the model, and objective of this exercise was to explore for physical significance in the ANN rainfall-runoff models of black-box type. The development of both BANN-BP and CRR Model–II models and their performance has already been discussed in detail in Chapter 4 and Chapter 5 respectively. It is anticipated that the scientific views and analyses presented in this study may initiate further research in the area of exploration of the physical significance inside ANN hydrological models.

6.2 Analysis for Exploration of Physical Significance in ANN Models

As mentioned earlier, there are many techniques that can be employed to understand the physical behavior of an ANN model, such as sensitivity analysis, correlation analysis, graphical techniques, and symbolic rules. In this study, correlation analysis and graphical techniques have been employed for the purpose of the exploration of physical significance in ANN rainfall-runoff models. The concepts, procedures used, and the

methodologies implemented for this purposes are explained in details in the following paragraphs.

An ANN consists of a massively parallel structure through which the information presented at the input layer is processed in the forward direction to compute the overall output as transformation of the summation through a non-linear logistic function. During the feed-forward calculations, one determines the output vector at each hidden neuron correspondingly. Let us call it [*HD*], with an element represented by *Hi*, $i = 1$ to n_1, where i is an index representing the number of neurons in the hidden layer, and n_1 is the total number of neurons in the hidden layer. Please note that the dimensions of the vector [*HD*] will be $n \times n_1$, where n is the total number of patterns in the training data set under consideration. Therefore, each *Hi* represents a 1-D vector consisting of outputs from the i^{th} neuron in the hidden layer corresponding to all patterns. The output from each hidden neuron [*Hi*] can be correlated with either the individual input vectors or the components of the CRR model for the same training data set.

The nodal responses of each hidden neuron of the trained ANN model were computed using Equation. (3.1). The strength of the relationship between these responses and the individual input variable as well as the conceptual components of the CRR model was then measured by computing the Pearson cross correlation coefficient. The discussion of exploring the ANN rainfall-runoff model for physical significance inherent in it has been systematically organized into two parts. The first part examines the relationships of the distributed components in the ANN rainfall-runoff model in terms of responses from various hidden neurons with the observed input vector employed in the ANN. The second part examines the strength of correlation between the responses at various

hidden neurons and the conceptual elements of the hydrologic process computed in the CRR model. Such a systematic approach is necessary in a study such as this because the examination of the results obtained in the first part can greatly facilitate the analyses and interpretation of the results obtained in the second part.

6.2.1 Hidden Neurons with Observed Input Variables

The structure of the final BANN-BP model consists of five neurons in the input layer, four neurons in the hidden layer, and one neuron in the output layer (i.e. 5-4-1). The input neurons represented P(t), P(t-1), P(t-2), Q(t-1), and Q(t-2) and the output neuron represents the streamflow at time t, Q(t), being modeled. Let the four hidden neurons be represented by H1, H2, H3, and H4. The training data set of 13 years (1960-1972) was used to compute the cross-correlation coefficients among outputs from H1, H2, H3, and H4 and other physical variables.

The cross-correlation coefficients between the hidden neuron outputs (H1, H2, H3, and H4) and the observed input variables are shown in Table 6.1. The outputs from hidden neurons H1, H2, and H4 are strongly correlated with observed flows in the past, Q(t-1) and Q(t-2), with correlation coefficient values (r) ranging from -0.6922 to -0.8228. Further, the relationships of both H1 and H2 are stronger with Q(t-1) as compared to those with Q(t-2), and the trend is opposite in case of H4. The strong correlations of neurons H1, H2, and H4 with past observed flows indicate that these neurons may be modeling either base flow or the surface flow component of the rainfall-runoff because the past observed flow is a combination of surface and base flow. The examination of the strength of correlation of these hidden neurons with the other key conceptual components computed in the CRR Model-II is needed to evaluate the effect

Table 6.1: Correlation Statistics of the Hidden Neurons and Observed Input Variables

Input Variables	Hidden Nodes			
	H1	**H2**	**H3**	**H4**
P(t-2)	-0.2765	-0.4454	-0.5372	0.0782
P(t-1)	-0.2022	-0.1477	-0.7378	-0.1305
P(t)	-0.1336	0.0208	-0.5277	-0.4487
Q(t-2)	-0.6922	-0.7050	-0.0467	-0.8058
Q(t-1)	-0.7079	-0.8228	-0.3295	-0.7835

of an individual hidden neuron on a particular hydrologic process. The direction of correlation coefficient represented by the sign (positive/negative) of the correlation coefficient is insignificant in the context of current analysis as the analysis deliberates the quantitative strength of relationship only and not the direction.

The neuron H3 seems to be correlated with the present and past rainfall with its dependence on previous day rainfall, P(t-1), being the strongest (r = -0.7378). It may be noted that among all the hidden neurons, only H3 is correlated strongly with the observed rainfall in the past and present. The relationships of H1, H2, and H4 with rainfall amounts on various days, range from very poor to only marginal with correlation coefficient values ranging from 0.0208 to 0.4487. The strong correlation of H3 with rainfall indicates that H3 may be modeling the effective rainfall or infiltration process. This needs to be further investigated by examining the strength of correlation of H3 with infiltration amounts computed from the CRR Model-II, and is discussed in the following section. It is worth mentioning that the cross-correlation between the hidden neurons themselves indicated that the four hidden neurons are not correlated to each other at all suggesting that they are independent of each other.

6.2.2 Hidden Neurons with Conceptual Components of Hydrologic Process

The cross-correlation coefficients between the hidden neuron outputs (H1, H2, H3, and H4) and the conceptual components of the hydrologic processes computed from the CRR Model-II are shown in Table 6.2. The Conceptual components considered in this study include base flow at time t, QG(t), surface flow at time t, QS(t), soil moisture content at time t, SM(t), actual incremental infiltration during a day t, ΔF(t), and the total computed flow from the CRR model, QC(t). It can be observed from Table 6.2

Table 6.2: Correlation Statistics of the Hidden Neurons and Conceptual Components of the Hydrologic Process

Deterministic Component	Hidden Nodes			
	H1	H2	H3	H4
QC(t)	-0.6712	-0.8011	-0.4840	-0.7314
QG(t)	-0.6266	-0.2011	-0.1275	-0.5268
QS(t)	-0.5653	-0.7830	-0.5639	-0.6589
SM(t)	-0.6547	-0.2445	-0.0902	-0.4903
ΔF(t)	-0.1055	0.0371	-0.4435	-0.3447

that all the hidden neuron responses are reasonably correlated with the output computed from the CRR model (r = -0.4840 to -0.8011) suggesting that the four hidden neurons may be representing certain components of the complex rainfall-runoff process as QC(t) is computed considering the physics of the hydrologic process. The output from neuron H1 is correlated well with base flow and soil moisture content with the correlation coefficient values of 0.6266 and 0.6547, respectively. Further, as stated earlier, neuron H1 is not correlated with either the past or present rainfall amounts, or the actual incremental infiltration component (r = -0.1055), suggesting that the hidden neuron H1 is modeling the base flow component of the rainfall-runoff process. The outputs from hidden neurons H2 and H4 are strongly correlated with the surface flow component QS(t) with correlation coefficients values of -0.7830 and -0.6589, respectively indicating that the hidden neurons H2 and H4 model the surface flow component of the rainfall-runoff process. Examining further, it can be noticed that H4 is more strongly correlated with the soil moisture component, SM(t), as compared to H2 (r = -0.4903 for H4 versus r = -0.2445 only for H2) indicating that H4 represents the quick flow portion (or interflow) and H2 represents the delayed flow component of the surface flow. The correlation of the actual incremental infiltration, $\Delta F(t)$, with the output from hidden neuron H3 is the strongest (r = -0.4437) suggesting that the hidden neuron H3 is helping the ANN to model the infiltration component of the rainfall-runoff process because H3 is the only hidden neuron strongly correlated with the observed rainfall in the past and present (see Table 6.1). This inference is further augmented by the fact that H3 is the only neuron that was found to be reasonably correlated with effective rainfall (r = -0.490).

The foregoing analyses suggest that it is possible to attribute physical significance to the hidden neurons of an ANN rainfall-runoff model if the massively parallel structure of the ANN is explored. The results obtained in this study suggest that a hidden neuron, which is strongly correlated with base flow and not correlated with past or present rainfall, is able to represent the final portion of the falling limb of the flow hydrograph. A hidden neuron, which is strongly correlated with surface flow and not so strongly related to either soil moisture or the infiltration models the delayed surface flow and represents the middle portion of the falling limb of the flow hydrograph. A hidden neuron, which is correlated with the effective rainfalls and infiltration amounts, is able to model the rising limb of the flow hydrograph because the rising limb of a flow hydrograph is dominated by the infiltration capacities and the rainfall characteristics such as intensity and duration of the rainfall and hence effective rainfall (Zhang and Govindaraju, 2000). Finally, a hidden neuron, which is strongly correlated with surface flow and the soil moisture in the watershed, models the interflow and represents the initial portion of the falling limb of the flow hydrograph just after the peak.

The conclusions drawn above are based on the computed strengths of correlation between the responses from the hidden neurons and the coceptual elements of the hydrologic process computed in a CRR model along with the input variables. However, it may be noted that the coefficient of correlation represents only the 'linear dependence' between the two variables under consideration. A further analysis of results is needed that examines these relationships graphically so that any nonlinear relationship between the two can also be assessed. The scatter plots of the four hidden neuron responses against the corresponding deterministic elements of the hydrologic process are shown in Figure 6.1 through Figure 6.4. Specifically, Figure 6.1 shows the

scatter plot between H1 response and base flow, Figure 6.2 shows the scatter plot between H2 response and surface flow, Figure 6.3 shows the scatter plot between H3 response and infiltration, and Figure 6.4 shows the scatter plot between H4 response and surface flow, respectively.

It is clear from Figure 6.1 that the relationship between H1 response and base flow is certainly not linear and is non-linear in nature. The non-linear relationship does not seem to be very well defined but the hidden neuron H1 seems to envelop the upper limit of the relationship, which is of the form of exponential decay normally apparent in the base flow portion of the falling limb. The non-linear relationship between H1 response and base flow would certainly be stronger than the linear dependence between them represented by r = -0.6266. This graphical representation strengthen the inference that hidden neuron H1 models the base flow and represents final portions of the falling limb of the hydrograph. Figure 6.2 suggests that the relationship between H2 response and the surface flow is also non-linear and the strength of this non-linear relationship will certainly be more than that represented in the linear correlation coefficient. A clear non-linear trend between H2 response and QS(t) further justifies the conclusion that the hidden neuron H2 models delayed surface flow and represents the middle portions of the falling limb of the hydrograph. The relationship between H4 response and surface flow, depicted in Figure 6.4, also suggests that the relationship is non-linear in nature and the strength of this non-linear relationship will also be better than a linear relationship revealed in the correlation coefficient. This justifies the conclusion that hidden neuron H4 models interflow and represents the initial portions of the falling limb just after the peak. Another interesting point worth mentioning here is that the hidden neurons H2 and H4 model the delayed surface flow and quick surface flow (interflow),

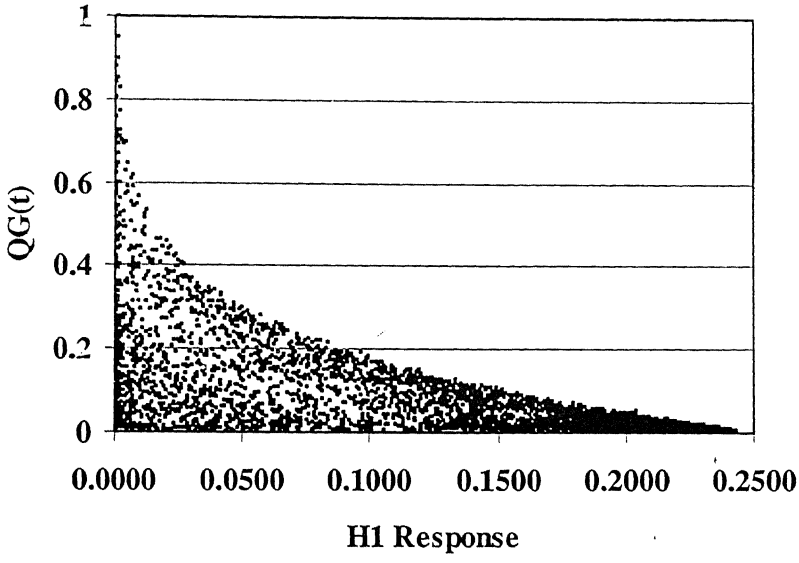


Figure 6.1: Scatter Plot: H1 Response v/s Base Flow

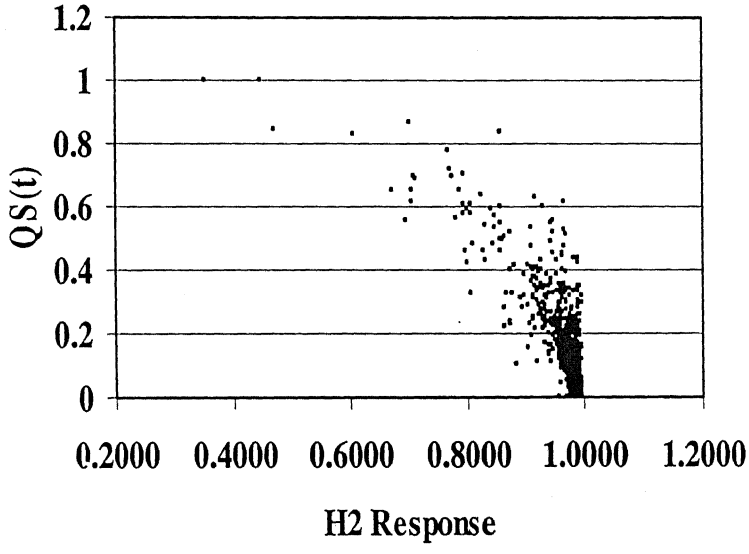


Figure 6.2: Scatter Plot: H2 Response v/s Surface Flow

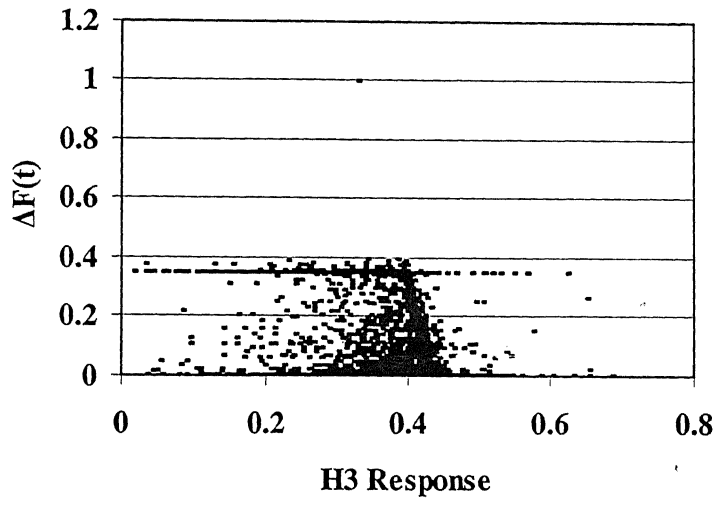


Figure 6.3: Scatter Plot: H3 Response v/s Infiltration

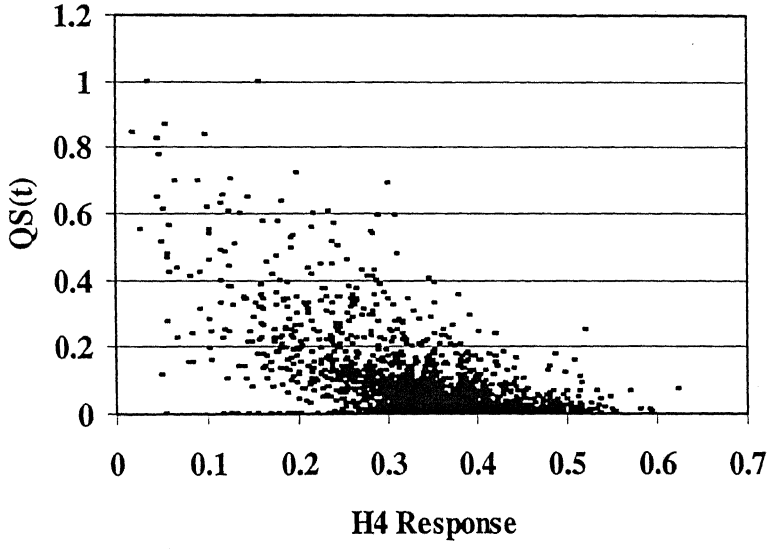


Figure 6.4: Scatter Plot: H4 Response v/s Surface Flow

respectively, which compliment each other. In other words, at any give point of time in a watershed, higher magnitudes of interflow would correspond to the lower magnitudes of the delayed surface flow; and lower magnitudes of interflow would coincide with higher magnitudes of delayed surface flow. Such complementary nature of interflow and delayed surface flows can be verified by Figure 6.2 in which the non-linear relationship is concave and in Figure 6.4 in which the non-linear relationship is convex in nature, which essentially means that higher magnitudes of interflow corresponds to the lower magnitudes of delayed surface flow and vice-versa. The relationship between H3 response and the infiltration amount does not seem to have any definite trend, also represented by r = -0.4435. The not so strong correlation between H3 response and the infiltration amount can probably be attributed to the errors in the rainfall measurements. It is worth mentioning that the rainfall data can be more erroneous than the streamflow that is normally more reliable. However, it can be noted that the relationship between H3 response and infiltration is definitely not linear, and there is a possibility of construing one or more non-linear relationships in different magnitudes, barring the outliers.

The conclusions drawn about the different hidden neurons representing different components of the hydrologic process is further augmented by the ranges of the magnitudes of the hidden neuron responses. The magnitude of the hidden neuron is actually the output from the sigmoid function, therefore, magnitude close to one represents high flows; whereas, the magnitudes close to zero would represent low flows. It can be noticed from Figure 6.1 that the magnitudes of H1 responses range from 0.0 to 0.25 only suggesting that H1 response is limited to a maximum of 0.25, which resembles the low magnitudes of the base flow it represents. The H2 responses range

from 0.4 to 1.0, and H4 responses range from 0.0 to 0.6 approximately. Both H2 and H4 are correlated well with the surface flow, which normally can have magnitudes in the entire flow range as verified by the combined magnitudes of H2 and H4 responses. Since H3 represents the infiltration process, a greater concentration of scatter points close to zero can be seen from Figure 6.3.

In summary, it is clear that if the structure of a trained ANN is examined closely with respect to the components of a physical process being modeled, it is possible to find resemblance of physics inside an ANN. The evidence that distributed components of an ANN are able to model or represent different components of a physical process contradicts the general notion that ANNs are purely black box models and do not contain the underlying physical process being modeled. This study has probably provided a platform to initiate a long-term debate about whether ANNs are purely black box models or do represent the physical processes in its architecture.

Chapter 7

Summary, Conclusions and Limitations

7.1 Summary

This thesis presents the findings of a study carried out with the purpose of developing improved methodologies for modeling the complex, dynamic, non-linear, and fragmented rainfall-runoff process in a large watershed. A variety of techniques ranging from simple conceptual techniques to the soft computing techniques, and their combinations were explored in an attempt to develop rainfall-runoff models that can be more efficient and effective in producing accurate runoff forecast, important inputs to many water resources planning, development, design, operation, and management systems. The techniques investigated include conceptual techniques for developing conceptual rainfall-runoff (CRR) models, artificial neural network (ANN) technique for developing ANN rainfall-runoff models, *elitist real-coded* genetic algorithm (RGA) for calibration of CRR models and training of the ANN models, and physics based heuristic and soft decomposition techniques for decomposing input output data space into different classes corresponding to different dynamics of rainfall-runoff process.

A new class of models is proposed that is capable of exploiting the advantages of both conceptual and non-linear systems theoretic technique of ANNs called *grey-box* models. The *grey-box* models are capable of embedding component(s) of the hydrologic process in an overall ANN modeling framework such that the details of the underlying physics

are visible through the overall mathematical model, though in a partial sense. Specifically, two CRR models, two black box type ANN models (termed BANN models), two *grey-box* type ANN models (termed GANN models), five *grey-box* ANN models employing heuristic decomposition technique (termed GDANN models), and two *grey-box* ANN models employing soft decomposition technique of self organization map (SOM) network classifiers were developed in this study. The two CRR models were similar in structures in modeling the base flow, infiltration, soil moisture accounting, and evapotranspiration but differed in the manner in which the surface flow component was modeled. The CRR Model-I accounted for both translation and attenuation of the watershed by modeling the surface flow component by assuming the watershed to be a linear reservoir. The CRR Model-II used a linear channel element in the form of a time area diagram to account for the translation effects and accounted for the attenuation effects by routing the output from the linear channel element from a non-linear reservoir representing the watershed. The Green-Ampt infiltration equations were employed to compute infiltration, as they provide exact analytical solution to an approximate physically based model, and also allow the use of a mechanism to continuously update the soil moisture storage.

The black-box ANN models employed total rainfall and past flow values as the inputs; whereas, the GANN, GDANN, and SOM models employed effective rainfall and past flow as inputs. The first GDANN model (GDANN-I) decomposed the flow hydrograph into two parts, a rising limb and a falling limb, and then modeled each of them using two separate ANNs. The second GDANN model (GDANN-II) was the same as GDANN-I on the rising limb but modeled the falling limb using the concept of flow recession. The third GDANN model (GDANN-III) was the same as GDANN-II on the

rising limb, and further decomposed the falling limb into two parts. The fourth GDANN model (GDANN-IV) was the same as GDANN-III on the falling limb, and decomposed the rising limb also into two parts. The first four models (GDANN-I through GDANN-IV) mentioned above decomposed the data using heuristic technique based on the physics of the hydrological processes, and trained the ANN models using BP method. The fifth GDANN model (GDANN-V) was the same as GDANN-IV model except that the ANN components in the GDANN-V model were trained using *elitist* RGA unlike other GDANN models in which ANN components were trained using BP method.

The use of *elitist* RGA has been explored, probably for the first time, to train the ANN models of the complex, dynamic, non-linear, and fragmented rainfall-runoff process. The daily total rainfall (mm) and daily mean streamflow (m^3/s) data derived from the Kentucky River watershed, USA, were employed to develop all the models investigated in this study. The performance of the models investigated in this study was evaluated using eight different performance evaluation indices. Further, the thesis also makes an attempt to explore for the physical significance in the ANN rainfall-runoff models of black box type using correlation analysis and graphical techniques.

7.2 Conclusions

The performance of the two CRR models in terms of various performance evaluation indices was found to be reasonably good. However, as expected, the CRR Model-II was found to be superior to the CRR Model-I in modeling the complex, dynamic, and non-linear rainfall-runoff process in the Kentucky River watershed due to its better structure in modeling the surface flow component. The use of *elitist* RGA for

calibrating the CRR model parameters was found to be successful and can be particularly useful as it is easy to implement, and allows the use of continuous record of rainfall and streamflow for many years to be employed in calibration as opposed to using certain representative storms coupled with HEC-1's calibration capability.

The performance of the BANN models was found to be better than the CRR models in terms of efficiency in modeling but CRR models performed slightly better than the BANN models in terms of effectiveness in predicting flows. This was due to the poor generalization ability of the popular back-propagation (BP) algorithm employed in the BANN models for training the ANNs, especially the low magnitude flows, which was overcome by the use of *elitist* RGA in training the BANN models. The results obtained in this study indicate that the incorporation of conceptual components in the ANN models improves the overall efficiency and effectiveness of the ANN rainfall-runoff models marginally. The training of GANN models by the *elitist* RGA also showed significant improvements in the overall generalization capability of the ANNs in modeling the complex, dynamic, non-linear, and fragmented rainfall-runoff process.

The gradual decomposition of the flow hydrograph into different segments using the physics based heuristic decomposition and the use of separate techniques to model different segments of the flow hydrograph, was found to result in gradual improvements in both effectiveness in modeling and effectiveness in predicting flows accurately from GDANN models of the complex, dynamic, non-linear, and fragmented rainfall-runoff process. Decomposition of the input output data into different classes using the SOM network classifiers coupled with ANN models supports the concept that different dynamical processes inherent in different data sets belonging to different classes should

be modeled separately to get better model performance. Further, the superiority of GDANN-V model over the SOM models indicates that dividing rainfall-runoff data space into different classes corresponding to different dynamics based on conceptual methods is a better approach than the soft decomposition using SOM network classifiers.

A model that is efficient in modeling and effective in prediction is desirable from application point of view as the runoff forecasts obtained from a rainfall-runoff model are normally used for operational purposes. However, it was observed in this study that a compromise or a trade-off had to be reached as it was found that the models that are excellent in efficient modeling were not very effective in prediction and vice versa. However, the *grey-box* rainfall-runoff model that employed physics based decomposition technique, and used *elitist* RGA for training the ANN component (GDANN-V model) was excellent in efficiency in modeling and very good in effectiveness in prediction and was deemed to be the best model among all the models structures investigated in this study. The unbiased standard statistical performance evaluation indices, e.g. average absolute relative error in prediction and threshold statistics coupled with the other performance evaluation indices normally employed such as R, E, NRMSE, NMBE, *etc.* must be employed to make better assessment and comparison of different models to select the best model to be employed in various water resources management activities rather than relying on NRMSE, R, E, NMBE, *etc.* that may be biased towards high flows. Further, it was observed that rather than using performance evaluation indices as global statistics for the whole data set, their use in assessing the effectiveness and efficiency of models in estimating varying magnitudes

of flows (such as low, medium, and high flows) using a partitioning analysis is more appropriate.

Moreover, many researchers have reported about the problems associated in modeling the low magnitude flows while developing ANN rainfall-runoff models trained using BP algorithm and its variations. This study presented two different improved methodologies for overcoming the problems of modeling low magnitude flows while developing ANN models of the complex, dynamic, non-linear, and fragmented rainfall-runoff process. The first methodology employed the *elitist* RGA to train the ANN rainfall-runoff models, which resulted in significant improvements in efficiency in modeling and estimation of low magnitude flows. The second methodology integrated the conceptual technique of flow recession to model the falling limb with the ANN technique to model the rising limb of the flow hydrograph to successfully overcome the problems associated in modeling the low magnitude flows. However, modeling the high and medium magnitude flows was found to be equally comparable in performance in terms of both efficiency and effectiveness from all the ANN models investigated in this study trained by either BP method or the *elitist* RGA.

The preliminary study exploring for the physical significance in the ANN rainfall-runoff models of the black box type revealed that it is possible to identify physical processes inherent in the massively parallel distributed nature of the ANN structure. The black box ANN model structure explored for physical significance in this study consisted of four neurons in the hidden layer. The results obtained in this study suggest that one hidden neuron was strongly correlated to base flow and was able to represent the final portion of the falling limb of the flow hydrograph. Another hidden neuron modeled the

delayed surface flow and represented the middle portion of the falling limb of the flow hydrograph. Another hidden neuron was found to be correlated with the effective rainfalls and infiltration, and was able to model the rising limb of the flow hydrograph. Finally, the fourth hidden neuron was found to be strongly correlated to surface flow and the soil moisture in the watershed, hence was able to model the interflow and represented the initial portion of the falling limb of the flow hydrograph just after the peak. The evidence of the physical significance inside the trained ANN rainfall-runoff models found in this study can potentially initiate a long-drawn debate on whether the ANN models should be considered as black box models.

7.3 Limitations and Scope for Further Work

No study is complete in itself and there are always limitations and scope for further improvements. In the light of the present research effort, the following limitations and scope for further research are identified.

- The methodologies presented in this thesis employed conceptual and ANN rainfall-runoff models developed using spatially aggregated rainfall taken from five different raingauges scattered throughout a large watershed. Spatial averaging of distributed rainfall tends to dampen the dynamic effects inherent in the rainfall-runoff relationship that is distributed in nature. Ideally, a distributed hydrologic model and an ANN rainfall-runoff model that employs individual rainfalls from different raingauges scattered throughout a large watershed need to be developed.
- The improvements in the performance of the *grey-box* models over the black box models were found to be only marginal. This may have been due to the

efficiency of the conceptual modeling techniques employed for computing effective rainfalls before presenting the same to the input layer of the *grey-box* ANN models, or measurement errors in the data not being handled properly by the conceptual methods. Though the Green Ampt infiltration equations were employed to model the infiltration process, it may be possible to further improve the performance of *grey-box* ANN models by using some other analytical infiltration model. Also, using convoluted effective rainfalls which have the translation effects of a watershed embedded in them may provide a better alternative for use in the input vector of the ANN models. However, these issues need to be investigated further.

- Another limitation of the work is the use of Haan's method of computing daily expected evapotranspiration in CRR and *grey-box* ANN models. It may be possible to achieve improved performance in rainfall-runoff modeling in a large watershed if another conceptual or empirical method of evapotranspiration estimation, such as Penman's method, is employed. However, a major drawback of the Penman's method is that it requires a lot of hydrological and climatic data that are not easy to obtain and it is difficult to implement. Nevertheless, the use of such methods are expected to provide better overall model performance but it needs to be investigated.
- The output from all the ANN models investigated in this study consisted of the daily flow at time t, Q(t). It would be interesting to use the surface flow, QS(t), at the output layer in the ANN rainfall runoff models, as this would increase the degree of conceptualism embedded in the *grey-box* ANN models and hence may result in better model performance in modeling the complex, dynamic, and non-linear rainfall runoff process.

- It was observed in this study that though the use *elitist* RGA for training improved the performance of all the ANN models in both efficiency and effectiveness, especially in estimating the low magnitude flows resulting in better overall generalized rainfall-runoff relationships, the use of *elitist* RGA is computationally expensive as compared to the BP algorithm. However, when the need of accurate runoff forecasts to be employed in important water resources systems planning, design, operation, and management projects is concerned, the computational effort may be justified. Further, once the ANN rainfall-runoff model has been trained using the *elitist* RGA method, the procedure of using the model in forecasting the runoff is same regardless of the training method employed. Further, the use of an ANN rainfall-runoff model to predict runoff is much simpler in use and implementation than the use of a complex CRR model. Therefore, instead of attempting to develop a more complex CRR model in order to achieve better forecast accuracy, the use of GDANN models may be justified, which is easier to implement.
- The conclusions drawn from the preliminary study on the exploration of physical processes in the ANN rainfall-runoff models are based on a single case study, and need to be thoroughly explored and reinforced through further rigorous research by carrying out similar studies in other watersheds of varying hydrologic and climatic conditions. Further, this study considers only the massively parallel distributed structure of the ANN to explore physics embedded in it, and does not closely examine the optimized weights directly. The optimized weights of an ANN, that are similar to the parameters of a mathematical model, contain important information related to the physical behavior and characteristics of the watershed. This is another area which

remains to be investigated in the pursuit of establishing the ANNs to be the most effective and powerful tools for modeling the complex, dynamic, non-linear, and fragmented rainfall-runoff process.

It is hoped that this thesis has provided a platform from where to launch an extensive research initiative focusing on the development of improved methodologies for rainfall-runoff modeling using integration of conceptual, soft computing, decomposition, and other techniques to improve the accuracy of the runoff forecasts to be employed in important water resources activities. It is also hoped that the study carried out in this thesis on the exploration of physical significance in ANN rainfall-runoff models will go a long way in removing the reluctance in their use and establishing the ANNs to be the most powerful tools of modeling and forecasting the water resources systems.

REFERENCES

(1) Abbott, M.B., Bathurst, J.C., Cunge, J.A., O'Connell, P.E. and Rasmussen, J. (1986), " An introduction to the European Hydrological System–Système Hydrologique Européen 'SHE'", Journal of Hydrology, 87: 45-77.

(2) Abrahart, R.J., See, L. and Kneale, P.E. (2001), " Investigating the Role of Saliency with a Neural Network Rainfall-Runoff Model", Computers & Geosciences, 27: 921-928.

(3) Amri, S. –i., Murata, N., Müller, K.-R., Finke, M. and Yang, H.H. (1997), " Asymptotic Statistical Theory of Over Training and Cross-Validation", IEEE Transactions on Neural Networks, 8(5), 985-996.

(4) ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000a), "Artificial Neural Networks in Hydrology I: Preliminary Concepts", Journal of Hydrologic Engineering, ASCE, 5 (2): 115-123.

(5) ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000b), "Artificial Neural Networks in Hydrology II: Hydrologic Applications", Journal of Hydrologic Engineering, ASCE, 5 (2): 124-137.

(6) Bäck, T., Fogel, D. and Michalewicz, Z. (Eds) (1997). Handbook of Evolutionary Computation, Bristol: Institute of Physics Publishing and New York: Oxford University Press.

(7) Battiti, B (1992). "First and Second Order Methods for Learning: Between Steepest Descent and Newton's Method", Neural Computation, 4(2): 141-166.

(8) Benitez, J. M., Castro, J. L., and Requena, I. (1997), "Are Artificial Neural Networks Black Boxes?", IEEE Transactions on Neural Networks, 8(5): 1156-1164.

(9) Bergström, S. (1995), "The HBV model. In Computer models in watershed hydrology", V.P. Singh (Ed.), Water Resources Publications: 443-476.

(10) Beven, K.J., Calver, A. and Morris, E.M. (1987), "The Institute of Hydrology Distributed Model", Institute of Hydrology, Report no. 8, Wallingford (UK), 33p.

(11) Beven, K.J. (1997), "TOPMODEL: A Critique", Hydrological Processes, 11(9): 1069-1085.

(12) Birikundavyi, S., Labib, R., Trung, H.T. and Rousselle, J. (2002), "Performance of Neural Networks in Daily Streamflow Forecasting", Journal Hydrologic Engineering, ASCE, 7(5): 392-398.

(13) Bishop, C. M. (1995), Neural Networks for Pattern Recognition, Clarendon, Oxford, UK.

(14) Brazil, L.E. and Krajewski, W.F. (1987), " Optimization of Complex Hydrologic Models using Random Search Methods", Proceedings of ASCE Conference on Engineering Hydrology, Williamsburg, New York, 726-731.

(15) Campolo, M., Andreussi, P. and Soldati, A. (1999), "River Flood Forecasting with Neural Network Model", Water Resources Research, 35(4): 1191-1197.

(16) Castro, J. L., Mantas, C. J. and Benitez, J. M. (2002), "Interpretation of artificial neural networks by means of fuzzy rules", IEEE Transactions on Neural Networks, 13(1):101-116.

(17) Cheng, B. and Titterington, D.M. (1994), "Neural Networks: A Review from Statistical Perspective", Statistical Science, 9(1): 2-54.

(18) Chiew, F.H.S. and McMahon, T.A. (1994), "Application of the Daily Rainfall-Runoff Model MODHYDROLOG to 28 Australian Catchments", Journal of Hydrology, 153: 383-416.

(19) Crawford, N.H. and Linsley, R. K. (1966), "Digital Simulation in Hydrology: Stanford Watershed Model IV", Technical Report 39, Department of Civil Engineering, Stanford University.

(20) Crespo, J.L., Mora, E. (1993), "Drought Estimation with Neural Networks", Advances in Engineering Software, 18(3): 167-170.

(21) Curry, B. and Morgan, P. (1997), "Neural Network: A Need for Caution", Omega, International Journal of Management Science, 25(1): 123-133.

(22) Cybenko, G. (1989), "Approximation by Superposition of a Sigmoidal Function", Mathematical Control Signal Systems, 2: 303-314.

(23) Dawson, D.W. and Wilby, R.L. (1998), "An Artificial Neural Network Approach to Rainfall-Runoff Modeling", Hydrological Sciences - Journal, 43 (1): 47-65.

(24) Dawson, D.W. and Wilby, R.L. (2001), "Hydrological Modeling using Artificial Neural Networks", Progress in Physical Geography, 25 (1): 80-108.

(25) Deb, K. and Agarwal, R.B. (1995), "Simulated Binary Crossover for Continuous Search Space", Complex Systems, 9: 115-148.

(26) Deb, K and Goyal, M. (1996), "A Combined Genetic Adaptive Search (GeneAS) for Engineering Design", Computer Science and Informatics, 26(4): 30-45.

(27) Deb, K. (2000), "An Efficient Constraint Handling Method for Genetic Algorithms", Computer Methods in Applied Mechanical Engineering, 186(2-4): 311-338.

(28) Deb. K. (2001), Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Ltd. Chichester.

(29) Dooge, J.C.I. (1959), " A General Theory of the Unit Hydrograph", Journal of Geophysical Research, 64(1): 241.

(30) Duan, Q.Y., Gupta, V.K., Sorooshian, S. (1992), "Effective and Efficient Global Optimization for Conceptual Rainfall-Runoff Models", Water Resources Research, 28(4): 1015-1031.

(31) Duan, Q.Y., Gupta, V.K., Sorooshian, S. (1993), "Shuffled Complex Evolution Approach for Effective and Efficient Global Optimization", Journal of Optimization: Theory & Applications, 76(3): 501-521.

(32) Ehrman, J.M., Higuchi, K., and Clair, T.A. (2000), "Backcasting to Test the use of Neural Networks", Canadian Water Resources Journal, 25(3): 279-291.

(33) Elshorbagy, A., Simonovic, S.P. and Panu, U.S. (2000), "Performance Evaluation of Artificial Neural Networks for Runoff Prediction", Journal Hydrologic Engineering, ASCE, 5(4): 424-427.

(34) Elshorbagy, A., Simonovic, S.P., and Panu, U.S. (2002), "Estimation of Missing Streamflow Data using Principles of Chaos Theory", Journal of Hydrology, 255: 123-133.

(35) Edijanto, Nascimento, N.D.O., Yang , Makhlouf Z. and Michel, C. (1999), "GR3J: A Daily Watershed Model with Three Free Parameters", Hydrological Sciences - Journal., 44(2): 263-277.

(36) Fahlman, S. E., and Lebiere, C. (1990), " The cascade –correlation learning architecture."

(37) Advances in Neural Information Processing Systems 2, D. S. Touretzky, ed., Morgan Kaufmann, San Mateo, Calif.: 524-532.

(38) Fitch, J.P., Lehman, S. K., Dowla, F. U., Lu, S. K. (1991), " Ship Wake Detention Procedure using Conjugate Trained Artificial Neural Networks", IEEE transactions on Geoscinces And Remote Sensing, 29(5): 718-725.

(39) Flood, I. and Kartam, N. (1994 a), "Neural Networks in Civil Engineering I: Principles and Understating ", Journal of Computing in Civil Engineering, 8(2): 131-148,

(40) Flood, I. and Kartam, N. (1994 b), "Neural Networks in Civil Engineering II: Systems and Application ", Journal of Computing in Civil Engineering, 8(2): 149-162,

(41) Flood, I. and Kartam, N. (1997), "Systems ", in Artificial Neural Networks for Civil Engineers: Fundamentals and Applications, Kartam, N., Flood, I. and Jr. Garrett, J.H., Eds., ASCE, ch.2.

(42) Franchini, M. (1996), "Use of Genetic Algorithm Combined With a Local Search Method for the Automatic Calibration of Conceptual Rainfall-Runoff Models", Hydrological Sciences – Journal, 41(1), 21-39.

(43) Franchini, M. and Galeati, G. (1997), "Comparing Several Genetic Algorithm Schemes for the Calibration of Conceptual Rainfall-Runoff Models", Hydrological Sciences - Journal, 42(3): 357-379.

(44) Funahasi, K.-I. (1989), "On Approximate Realization of Continuous Mappings by Neural Networks", Neural Networks, 2(3): 183-192.

(45) Furundzic, D. (1998), "Application Example of Neural Networks for Time Series Analysis: Rainfall Runoff Modeling", Signal Processing, 64: 383-396.

(46) Gan, T. Y. and Biftu, G. F. (1996), "Automatic Calibration of Conceptual Rainfall-Runoff Models: Optimization Algorithms, Catchment Conditions, and Model Structure", Water Resources Research, 32(12): 3513-3524.

(47) Goldberg D.E., Deb, K. and Clark, J. H. (1992), Genetic Algorithms, Noise, and the Sizing of Populations", Complex Systems, 6(4): 333-362.

(48) Goyen, A.G. (1987), " RAFTS: Runoff Aanalysis and Flow Training Simulation (Version 2.5)", Detailed Documentation and Users Manual, W.P. Software, Curtin, ACT.

(49) Grayson, R.B., Moore, I.D., and McMahon, T.A. (1992), "Physically Based Hydrologic Modeling-2, Is the Concept Realistic", Water Resources Research, 28 (10): 2659-2666.

(50) Gupta, J.N.D. and Sexton, R.S. (1991), "Comparing Back-Propagation with a Genetic Algorithm for Neural Network Training", Omega, International Journal of Management Science, 27: 679-684.

(51) Haan, C.T. (1972), A Water Yield Model for Small Watersheds, Water Resources Research, 8 (1): 58-69.

(52) Herara, F., Lozano, M. and Verdegay, J.L. (1998), "Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioral Analysis", Artificial Intelligence Review, 12(4): 265-319.

(53) Hornik, K., Stinchcombe, M. and White, H. (1989), Multilayer Feedforward Networks are Universal Function Approximators", Neural Networks, 2(5), 359-366.

(54) Hecht-Nielsen, R. (1990), Neurocomputing , Addison-Wesley, Reading, Mass.

(55) Hendrickson, J., Sorooshian, S. and Brazil, L.E. (1988), "Comparison of Newton-Type and Direct Search Algorithms for Calibration of Conceptual Rainfall-Runoff Models", Water Resources Research, 24(5), 691-700.

(56) Hsu, K.-L., Gupta, H.V. and Sorooshian, S. (1995), "Artificial Neural Network Modeling of the Rainfall-Runoff Process." Water Resources Research, 31(10): 2517-2530.

(57) Hsu, S.M.P.E., Ni, C-F, and Hung, P-F. (2002) "Assessment of Three Infiltration Formulas based on Model Fitting on Richards Equation", Journal of Hydrologic Engineering., 373-379.

(58) Hydrologic Engineering Center, (1990), HEC-1 Flood Hydrograph Package User's Manual and Programmer's Manual, US Army Corps of Engrs., Davis, California, USA.

(59) Iyengar, S. S., Cho, E.C., and Phoha, Vir, V. (2002), "Foundations of Wavelet Networks and Applications", Chapman & Hall/CRC, CRC Press LLC, Florida.

(60) Jaceman, A. J., Littlewood, I.G. and Whitehead, P.G. (1990), "Computation of the Instantaneous Unit Hydrograph and Identifiable Component Flows with Application to Two Small Upland Catchments", Journal of Hydrology, 117: 275-300.

(61) Jain, A. and Indurthy, S.K.V.P. (2003) "Comparative Analysis of Event based Rainfall-Runoff Modeling Techniques-Deterministic, Statistical, and Artificial Neural Networks", Journal of Hydrologic Engineering, ASCE, 8 (2): 1-6.

(62) Jain, A., Varshney, A.K., and Joshi, U.C. (2001), "Short-Term Water Demand Forecast Modeling at IIT Kanpur using Artificial Neural Networks", Water Resources Management, 15(5): 299-321.

(63) Jain, A. and Ormsbee, L.E. (2002), "Evaluation of Short-Term Water Demand Forecast Modeling Techniques: Conventional Methods versus AI", Journal of American Water Works Association, 94(7): 64-72.

(64) Karunanithi, N., Grenney, W. J., Whitley, D. and Bovee, K. (1994), "Neural Networks for River Flow Prediction ", Journal of Computing in Civil Engineering, 8(2), 201-220.

(65) Kohonen, T. (1982), "Self-Organized Formation of Topologically Correct Feature Maps", Biological Cybernetics, 43: 59-69.

(66) Kolmogorov, A.N. (1957), "On the Representation of Continuous Functions of Several Variables by Super-Position of Continuous Variable and Addition", Doklady Akademii Nauk USSR, 114: 679-681.

(67) Kumar, A. and Minocha, K. (2001), "Discussion on Rainfall Runoff Modeling Using Artificial Neural Networks", Journal Hydrologic Engineering ASCE, 6(2), 176-177.

(68) Lakatos, D.F. (1976), " Analysis of the Timing of Subwatershed Response to Storms with Use of Computer Simulation Models", M.S. Thesis, Department of Civil Engineering, Penn State University, University Park, Pennsylvania.

(69) Lange, N. T. (1999), "New Mathematical Approaches in Hydrological Modeling-an Application of Artificial Neural Networks", Physics & Chemistry of Earth, 24 (1&2): 31-35.

(70) Littelewood, I.G., Down, K., Parker, J.R. and Post, D. A. (1997), The PC version of IHACRES for Catchment-Scale Rainfall-Stremflow Modeling. Version 1.0. User Guide. Institute of Hydrology, 89p.

(71) Lorrai, M,, and Sachi, G. M. (1995), "Neural Nets for Modeling Rainfall-Runoff Transformations" Water Resources Management, 9: 299-313.

(72) Lozowski, A., Cholewo, T. J. and Zurada, J. M. (1996), "Crisp rule extraction from perceptron network classifiers", Proceedings of IEEE International Conference on Neural Networks: Plenary, Panel and Special Sessions, Washington DC,94-99.

(73) Hassoun, M.H. (1999). Fundamentals of Artificial Neural Networks, Prentice–Hall of India Private Limited, New Delhi, India.

(74) Maier, H.R. and Dandy, G.C. (2000), "Neural Networks for the Prediction and Forecasting of Water Resources Variables: A Review of Modeling Issues and Applications", Environmental Modeling & Software, 15: 101-124.

(75) MacGregor, G. (2001), "Decomposition for Judgmental Forecasting and Estimation", in Principles of Forecasting: A Handbook for Researchers and Practitioners, Armstrong, J. S., Ed.,Kluwer Academic Publishers, London,107-123.

(76) McCulloch, W.C. and Pitts, W. (1943), " A Logical Calculus of the Ideas Immanent in Nervous Activity", Bulletin of Mathematical Biophysics, 5: 115-133.

(77) Metcalf and Eddy, Inc. (1971), "Storm Water Management Model", EPA Report 110224DOC.

(78) Michalewicz, Z. (1996), Genetic Algorithms + Data Structures = Evolution Programs, Springer. NY.

(79) Minns, A.W. and Hall, M.J. (1996), "Artificial Neural Networks as Rainfall Runoff Models", Hydrological Sciences - Journal, 41 (3): 399-417.

(80) Mizumura, K. (1995), "Runoff Prediction by Simple Tank Model using Recession Curves", Journal of Hydraulic Engineering, 121(11): 812-817.

(81) Moore, R.J. and Clarke, R.T. (1981), "A Distribution Function Approach to Rainfall Runoff Modeling", Water Resources Research, 17(5): 1367-1382.

(82) Motrl-Seytoux, H. J. and Al Hassoun, S. (1989), The Unsaturated Component of SWATCH: a Multiprocess watershed Model for Runoff Generation and Routing. In: H.J. Morel-Seytoux (Ed.) "Unsaturated Flow in Hydrologic Modeling – Theory and Practice", NATO ASI Series C, 275, Kluwer Academic Publications : 413-433.

(83) Mühlenbein, H., Schomisch, M. and Born, J. (1991), "the Parallel Genetic Algorithm as Function Optimizer", Parallel Computing, 17:619-632.

(84) Mulvany, T. J. (1850), "On the Use of Self-Registering Rain and Flood Gauges", Proceeding of Institution of Civil Engineers, 4(2): 1-8.

(85) Nash, J.E. (1957), "The form of Instantaneous Unit Hydrograph", Hydrological Sciences Bulletin, 3: 114-121.

(86) Nash, J.E. and Sutcliffe, J.V. (1970), "River Flow Forecasting through Conceptual Models Part I-A Discussion of Principles", Journal of Hydorlogy, 10: 282-290.

(87) Ndiritu, J. G. and Daniell, T.M. (2001), "An Improved Genetic Algorithm for Rainfall-Runoff Model Calibration and Function Optimization", Mathematical and Computer Modeling, 33: 695-706.

(88) Ooyen, A. V. and Nichhuis, B. (1992), "Improving Convergence of Back Propagation Problem", Neural Networks, 5: 465-471.

(89) Pickup, G. (1977), "Testing the Efficiencies of Algorithms and Strategies for Automatic Calibration of Rainfall-Runoff Models", Hydrological Sciences Bulletin, 22(2): 257-274.

(90) Perrin, C., Michel, C., and Andreassian, V. (2001), "Does a large number of parameters enhance model performance? Comparative Assessment of common

Catchment Model Structures on 429 Catchments", Journal of Hydrology, 242: 275-301.

(91) Porporato, A. and Ridolfi, L. (2001), "Multivariate Nonlinear Prediction of River Flows", Journal of Hydrology, 248: 109-122.

(92) Pronzato, L., Walter, E., Venot, A. and Lebruchec, J.-F. (1984), "A General Purpose Global Optimizer: Implementation and Applications", Mathematics and Computers in Simulation, 26:412-422.

(93) Rajurkar, M.P., Kothyari, U.C. and Chaube, U.C. (2002), "Artificial Neural Networks for Daily Rainfall-Runoff Modeling", Hydrological Sciences - Journal, 47(6): 865-876.

(94) Raman, H. and Sunil Kumar, N. (1995), "Multivariate Modeling of Water Resources Time Series Using Artificial Neural Networks", Hydrological Sciences - Journal, 40(2): 145-163.

(95) Refsagaard, J.C. and Kmudsen, J. (1996), "Operational Validation and Inter comparison of Different Types of Hydrological Models", Water Resources Research, 32(7): 2189-2202.

(96) Robert, J. A., Linda, S. and Pauline, E. K. (2001), "Investigating the Role of Saliency Analysis with a Neural Network Rainfall-Runoff Model", Computers and Geosciences, 27: 921-928.

(97) Rosenblatt, F. (1958), "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain", Psychological Review, 65: 386-408.

(98) Rosenblatt, F. (1961), Principles of Neurodynamics , Spartan Press, Washington, DC.

(99) Rosenbrock, H.H. (1960), "An Automatic Method of Fitting the Greatest or Least Value of a Function", Computer Journal, 3:175-184.

(100) Rudolph, G. (1996), "Convergence of Evolutionary Algorithms in General Search Spaces", In Proceedings of the Third IEEE Conference on Evolutionary Computation, 50-54.

(101) Rumelhart, D.E., Hinton, G.E. and Williams, R. J. (1986 a), "Learning Representations by Back-Propagating Errors", Nature, 323: 533-536.

(102) Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986 b), Learning Internal Representations by Error Back Propagation, in D.E. Rumelhart and J.L. McClelland (eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol 1, Foundations, The MIT Press, Ch 8.

(103) Safi, S., Zeroual, A., and Hassani, M. (2002), "Prediction of Global Daily Solar Radiation Using Higher Order Statistics", Renewable Energy, 27: 647-666.

(104) Sajikumar, N. and Thandaveswara, B.S. (1999), "A Non-Linear Rainfall-Runoff Model using an Artificial Neural Network", Journal of Hydrology, 216, 32-55.

(105) Schmitz, G. P. J., Aldrich, C., and Gouws, F. S. (1999), "ANN-DT: An algorithm for extraction of decision trees from artificial neural networks", IEEE Transactions on Neural Networks, 10(6): 1392-1401.

(106) Shamseldin, A. Y. (1997), "Application of a Neural Network Technique to Rainfall-Runoff Modeling", Journal of Hydrology, 199: 272-294.

(107) Sherman, L. K. (1932), "Streamflow from Rainfall by a Unit Hydrograph Method", Engineering News Record, 108: 501-505.

(108) Singh, V.P. (1988), "Hydrologic Systems: Vol 1", *Prentice Hall*, Englewood Cliffs, NJ.

(109) Sivakumar, B. (2001), "Is a chaotic multi-fractal approach for rainfall possible?", Hydrological Processes, 15: 943-955.

(110) Smith, J. and Eli, R. N. (1995), " Neural-Network Modeling of Rainfall-Runoff Process." Journal of Water Resources Planning and Management, 121(6): 499-507.

(111) Sorooshian,S., Gupta, V.K. (1983), " Automatic Calibration of Conceptual Rainfall-Runoff Models: The Question of Parameter Observability and Uniqueness", Water Resources Research, 19(1): 260-268.

(112) Sorooshian,S., Gupta, V.K. (1985), " The Analysis of Structural Identifiability: Theory and Application to Conceptual Rainfall-Runoff Models", Water Resources Research, 21(4): 487-495.

(113) Spreecher, D.A. (1993), "A Universal Mapping for a Kolmogorov's Superposition Theorem, Neural Networks, 6(8): 1089-1094.

(114) Stone, R.J. (1993), "Improved Statistical Procedure for the Evaluation of Solar Radiation Estimation Models", Solar Energy, 51(4): 289 – 291.

(115) Sugawara, M. (1995), "Tank model. In Computer models in watershed hydrology", V.P. Singh (Ed.), Water Resources Publications, 165-214.

(116) Subramanya, K. (1994), Engineering Hydrology, Tata McGraw Hill Publishing Company Ltd., New Delhi.

(117) Sudheer, K. P., Gosain, A. K., and Ramasastri, K. S. (2002), "A data-driven algorithm for constructing artificial neural network rainfall-runoff models." Hydrological Processes, 16: 1325-1330.

(118) Summer, N.R., Fleming, K.M., Bates, B.C. (1997), " Calibration of a modified SFB model for twenty-five Australian Catchments using Simulated Annealing", Journal of Hydrology, 197: 166-188.

(119) Tan, B.Q. and O'Connor, K.M. (1996), "Application of an Empirical Infiltration Equation in the SMAR Conceptual Model", Journal of Hydrology, 185: 275-295.

(120) Terstriep, M.L. and Stall, J.B. (1974), "The Illinois Urban Drainage Area Simulator, ILLUDAS", Illinois State Water Survey Bulletin 58.

(121) Thatte, C.D. (2002), "Water and Food Security", Keynote address at HYDRO2002, IIT Bombay.

(122) Thirumalaiah, K. and Deo, M.C. (2000), "Hydrological Forecasting Using Neural Networks", Journal of Hydrologic Engineering, ASCE, 5(2): 180-189.

(123) Thyer, M., Kuczera, G. and Bates, B.C. (1999), "Probabilistic Optimization for Conceptual Rainfall-Runoff Models: A Comparison of the Shuffled Complex Evolution and Simulated Annealing Algorithms", Water Resources Research, 35(3): 767-773.

(124) Tibshirani, R. (1994), "Comment on 'Neural Networks: A Review from a Statistical Perspective' by B. Cheng and D.M.Titterington", Statistical Science, 9(1): 48-49.

(125) Todini, E. (1996), "the ARNO rainfall-runoff model", Journal of Hydrology, 175, 339-382.

(126) Tokar, A.S. and Johnson, A. (1999), "Rainfall-Runoff Modeling Using Artificial Neural Networks." Journal Hydrologic Engineering, ASCE, 4(3): 232-239.

(127) Tokar, A. S., and Markus, M. (2000), "Precipitation Runoff Modeling Using Artificial Neural Network and Conceptual models." Journal Hydrologic Engineering, ASCE, 5(2): 156-161.

(128) Tollenaere, T. (1990), ":SuperSAB: Fast adaptive back propagation with good scaling properties", Neural Networks, 3: 561-573.

(129) Tomassini, M. (1993),"The Parallel Genetic Cellular Automata: Application to Global Function Optimization", In Artificial Neural Nets and Genetic Algorithms, (Edited by R.F. Albrecht et al.), 385-391: Srpinger-Verlag, New York.

(130) Tickle, A. B., Andrews, R., Golea, M. and Diederich, J. (1998), "The truth will come to light: Directions and challenges in extracting knowledge embedded within trained artificial neural network", IEEE Transactions on Neural Networks, 9(6), 1057-1068.

(131) Vitela, J.E. and Reifman, J. (1997), "Premature Saturation in Back Propagation Networks – Mechanism and Necessary Conditions", Neural Networks, 10(4): 721-735.

(132) Wang, Q.J. (1991), "The Genetic Algorithm and Its Application to Calibrating Conceptual Rainfall-Runoff Models." Water Resources Research, 27(9): 2467-2471.

(133) Wang, Q.J. (1997), "Using Genetic Algorithms to Optimize Model Parameters." Environment Modeling Software, 12(1): 27-34.

(134) Yapo, P.O., Gupta, H.S. and Sorooshian, S. (1996), "Automatic calibration of conceptual rainfall-runoff models: sensitivity to calibration data", Journal Hydrology 181, 23-48.

(135) Zhang, B. and Govindaraju, S. (2000), "Prediction of Watershed Runoff using Bayesian Concepts and Modular Neural Networks." Water Resources Research, 36 (3): 753-762.

(136) Zhao, R,-J. and Liu, X.-R. (1995), "The Xinanjiang model. In Computer models in watershed hydrology", V.P. Singh (Ed.), Water Resources Publications: 215-232.

(137) Zilochian, Z. and Jamshidi, M. (Eds) (2001), Intelligent Control Systems using Soft Computing Methodologies, CRC Press LLC.

BIBILOGRAPHY

(1) Anreassian, V., Perrin, C.,Michel, C., Usart-Sanchez, I., and Lavabre, J. (2001), "Impact Imperfect Rainfall Knowledge on the Efficiency and the Parameters of Watershed Models", Journal of Hydrology, 250 : 206-223.

(2) Arnaud, P., Bouvier, C., Cisneros, L. and Dominguez, R. (2002), "Influence of Rainfall Spatial Variability on Flood prediction", Journal of Hydrology, 260: 216-230.

(3) Basha, H. A. (2000), "Simple Nonlinear Rainfall-Runoff Model.", Journal of Hydrologic Engineering, 5(1): 25-32.

(4) Brown, M. and Harris, C. (1994), Neuro-fuzzy adaptive modeling and control, Prentice Hall, New York.

(5) Chiew, F.H.S., Stewardson, M.J. and McMahon, T.A. "Comparison of Six Rainfall-Runoff Modeling Approaches", Journal of Hydrology, 147: 1-36.

(6) Demaree, G. (1982), "Comparison of Techniques for the Optimization of Conceptual Hydrological Models", Mathematics and Computers in Simulation, XXIV : 122-130.

(7) Faures, J.-M,, Goodrich, D.C., Woolhiser, D.A. and Sorooshian, S. (1995), "Impact of Small-Scale Spatial Rainfall Variability on Runoff Modeling", Journal of Hydrology, 173 : 309-326.

(8) French, M.N., Krajewski, W.F. and Cuykendall, R. R. (1992), "Rainfall Forecasting in Space and Time using a Neural Network", Journal of Hydrology, 137 : 1-31.

(9) Goldberg, D.E. (2000), Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wiley, Bangalore, India.

(10) Houghton-Carr, H.A. (1999), "Assessment Criteria for Simple Conceptual Daily Rainfall – Runoff Models", Hydrological Sciences-Journal, 44(2): 237-261.

(11) Johnson, V.J. and Rogers, L.L. (2000), "Accuracy of Neural Network Approximators in Simulation-Optimization" Journal of Water Resources Planning and Management, 126(2): 48-56.

(12) Montana, D. J., and Davis, L. (1989), "Training Feed-forward Neural Networks Using Genetic Algorithms", Proc. of the 11th International Joint Conference on Artificial Intelligence, 1, 762-767.

(13) Kachroo, R.K. (1992), "River flow forecasting. Part 5. Applications of a conceptual model", Journal of Hydrology, 133: 141-178.

(14) Kachroo, R.K., Sea, C.H., Warsi, M.S., Jemenez, H. and Saxena, R.P. (1992.), " River flow forecasting. Part 3. Applications of Linear Techniques in Modeling Rainfall-Runoff Transformations", Journal of Hydrology, 133: 41-97.

(15) Muftuoglu, R.F. (1991), "Monthly Runoff Generation by Non-Linear Methods", Journal of Hydrology, 125: 277-291.

(16) Mason, J. C., Price, R.K. and Temm'me, A. (1996), "A Neural Network Model of Rainfall-Runoff using Radial Basis Functions", Journal of Hydraulic Research, 34: 537-548.

(17) Nalbantis, I. (1995), "Use of Multiple-Time-Step Information in Rainfall-Runoff Modeling", Journal of Hydrology, 165: 135-159.

(18) Paturel, J.E., Servat, E. and Vassiliadis, A. (1995), "Sensitivity of Conceptual Rainfall-Runoff Algorithms to Errors in Input Data–Case of the GR2M Model", Journal of Hydrology, 168: 111-125.

(19) Servat, E. and Dezetter, A. (1991), "Selection of Calibration Objective Functions in the Context of Rainfall-Runoff Modeling in a Sudanese Savannah Area", Hydrological Sciences-Journal, 36(4): 307-330.

(20) Tan, B.Q. and O'Connor, K.M. (1995), "Application of an Empirical Infiltration Equation in the SMAR Conceptual Model", Journal of Hydrology, 185: 275-295.

(21) Tsykin, E.N. (1985). "Multiple Nonlinear Statistical Models for Runoff Simulation and Prediction", Journal of Hydrology, 77: 209-226.

(22) Wilk, J. and Hughes, D.A. (2002), "Calibrating a Rainfall-Runoff Model for a Catchment with Limited Data", Hydrological Sciences-Journal, 47(1): 3-17.

(23) Zurada, J.M. (1999), Introduction to Artificial Neural Systems, Jaico Publishing House, Mubai, India.

APPENDIX-A

Rastrigin Function

$$f(x_1, x_2) = 2 + x_x^2 + x_2^2 - \cos(18x_1) - \cos(18x_2),$$

$$-1 \le x_1, x_2 \le 1$$

The global minimum is 0 at (0,0). There are more than 50 local minima in the region of interest, arranged in a lattice configuration.

Six-Hump Camelback Function

$$f(x_1, x_2) = 1.036285 + 4x_x^2 - 2.1x_1^4 + (1/3)x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4,$$

$$-2 \le x_1 \le 2,\ -1 \le x_2 \le 1,$$

this function systematic about the origin and has three pairs of local minima. The global minimum is 0 at (0.08983, -0.7126) and (-0.08983,0.7126).

Griewank Function

$$f(x) = \frac{\sum_{i=1}^{10} x_i^2}{4000} - \prod_{i=1}^{10} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

$$-600 \le x_i \le 600,\ i = 1, \ldots.10,$$

The global minimum is 0 and is at the origin. There are several thousand local minima in the region of interest.

Hartman Function

$$f(x) = 3.32 - \sum_{i=1}^{4} c_i \exp\left(- \sum_{j=1}^{6} \alpha_{i,j} (x_j - p_{i,j})^2 \right)$$

$$0 \le x_j \le 1, \; j = 1, \ldots, 6$$

The function have the global minimum is 0 at (0.201, 0.150, 0.477, 0.275, 0.311, 0.657). There are four local minima in the region of the interest. The values of the coefficients $\alpha_{i,j}$, c_i, $p_{i,j}$ as given below.

Table 1: Values of α_i and c_i for Hartman function

i	$\alpha_{1,i}$	$\alpha_{2,i}$	$\alpha_{3,i}$	$\alpha_{4,i}$	$\alpha_{5,i}$	$\alpha_{6,i}$	c_i
1	10.00	3.00	17.00	3.50	1.70	8.00	1.0
2	0.05	10.00	17.00	0.10	8.00	14.00	1.2
3	3.00	3.50	1.70	10.00	17.00	8.00	3.0
4	17.00	8.00	0.05	10.00	0.10	14.00	3.2

Table 2: Values of p_i for Hartman function

i	$p_{1,i}$	$p_{2,i}$	$p_{3,i}$	$p_{4,i}$	$p_{5,i}$	$p_{6,i}$
1	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650
4	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

Research Publications from this Study

International Journal

1. Discussion of Performance of Neural Networks in Daily Streamflow Forecasting. *J. Hydrol. Engg., ASCE* (2003, accepted)

2. Development of Effective and Efficient Rainfall-Runoff Models using Integration of Deterministic, Real-Coded GA, and ANN Techniques (*communicated Water Resources Research, 2003*)

3. Integrated Approach to Modeling Decomposed Flow Hydrograph using Artificial Neural Network and Deterministic Techniques, (*Communicated to Water Resources Research, 2003*).

4. Identification of Physical Processes inherent in Artificial Neural Network Rainfall-Runoff Models, (*Communicated to Hydrological* Processes)

International Conference

1. Determination of an Optimal Unit Hydrograph of River Basin using Genetic Algorithm. *2nd International conference on River Basin Management,, 28-30 April 2003*, Las Palmas, Gran Canaria, Spian.

2. Systems Theoretic Approach to Modeling Rainfall-Runoff Process with Conceptual Component, *2nd International conference on River Basin Management,, 28-30 April 2003*, Las Palmas, Gran Canaria, Spian.

3. Rainfall-Runoff Pattern Mapping using Artificial Neural Networks, *International Conference on Hydrology and Water shed management*, JNT University, Hyderabad. *December, 2002.*

4. Calibration of Infiltration Parameters using Real Coded Genetic Algorithm, *Conference on Conference on Hydraulics, Water Resources and Ocean Engineering "Hydro-2002"*, December 16-17, 2002, IIT Bombay, December 2002.